Spread spectrum direct sequence

Student: Daniel Kraus
Supervisor: Richard Guniee

TABLE OF CONTENT

1 Introduction

Before you start reading this report it is essential to read the article attached in appendix A, which presents good background theory information about spread spectrum techniques and terminology and I will use it as a reference in my report. As there were written so many reports and articles about spread spectrum, I really see no point in tying to write my own theory introduction and I would rather reprint quite good article by someone else and concentrate on my project report. Spread spectrum (SS later on) is one of the major trends in modern communication systems. First usage can be dated back to WWII, when it was used by military for secret and hard to jammed transmissions. SS excellent anti-jamming and secret properties then caused for long period of cold war, that the techniques of SS was kept secret only for military purposes. First commercial use was in 80's for satellite phones. Nowadays, SS is taking over in more and more applications. Just few to mention are: GPS – position location system, which uses clock synchronization from several satellites to calculate precise position on the globe, Blue-tooth, wireless LAN and other devices operating in 2.4GHz license-free band, They usually use frequency hoping to avoid interference. The last, but the most important example is 3-G cellular phone network, which is using SS direct sequence to achieve Code Division Multiple Access (CDMA), in other words simultaneous transmission of multiple transmitters at the same frequency, distinguished from each other only by different PN codes.

2 Project specification

The project I was assigned was basically to reproduce the design and to construct transmitter-receiver system using SS direct sequence based on the article by James Vincent enclosed in appendix A. After reading the article I reached with my supervisor the conclusion that the closer specification of my project will be as follows:

- System operating frequency will be only 70 MHz (first down conversion from 435MHz to 70MHz will be omitted as this would only cause serious problem in design and construction and will not bring any addition value to demonstration of SS functionality)
- Audio part will be omitted as well and there will not be any other circuitry to sample data in and out of the system, because again this prevent a lot of trouble with design and construction and has no additional value to project
- All other section will try to reproduce the original system, with possible minor redesigns.
- Signal generators will be used for carrier in transmitter and for local oscillator down conversion in receiver

3 Project achievements & results

I will start quite unusual with the project results at the beginning of the report instead of at the end as I think that the results are the only interesting figures of the project and only if you are interested in detailed functional description you can proceed to following topics.
I can sum up project results as following:

- I managed built functioning transmitter and receiver according to specification. Transmitter and receiver are on separate boards, and then I have another small board just for demonstration purposes. All 3 boards were mounted on prospect for better protection when moving boards (see Figure 1,2,3,4).

Figure 1: Whole system configuration

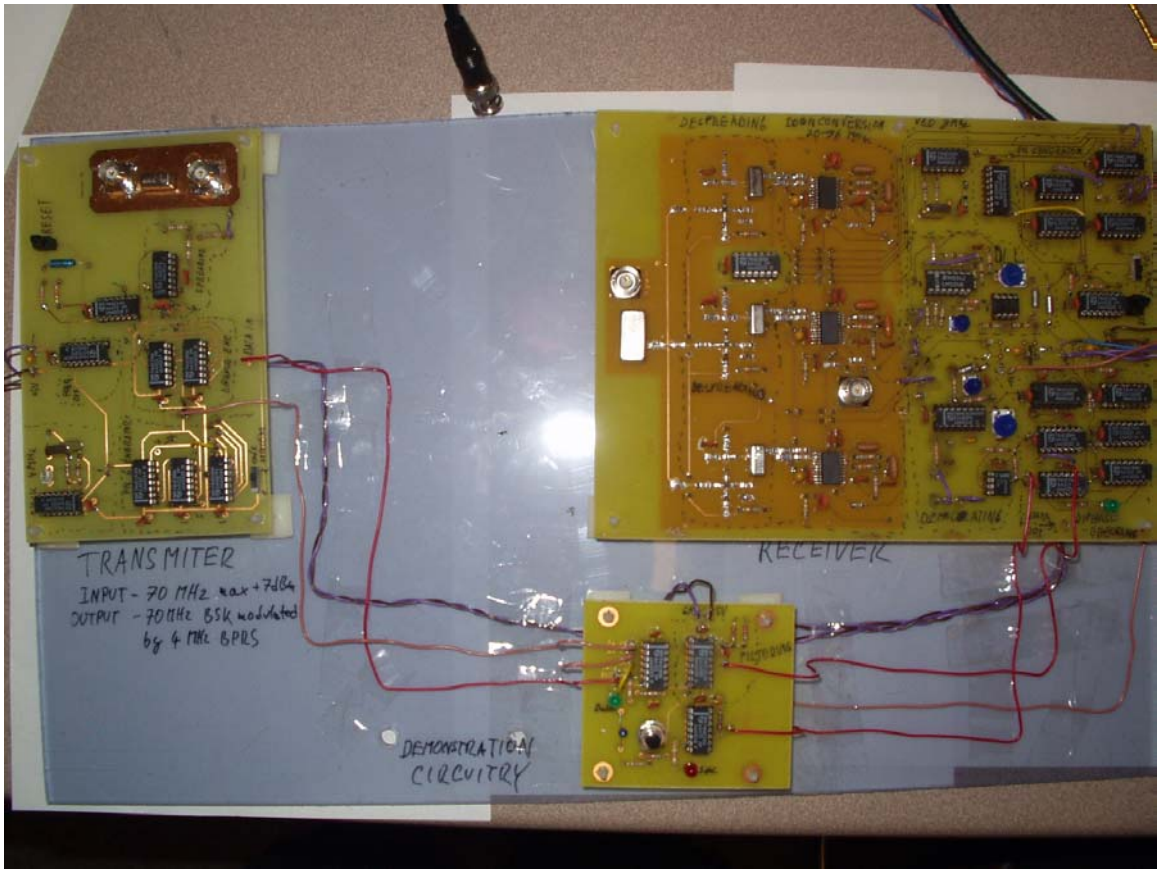


Figure 2: Transmitter module

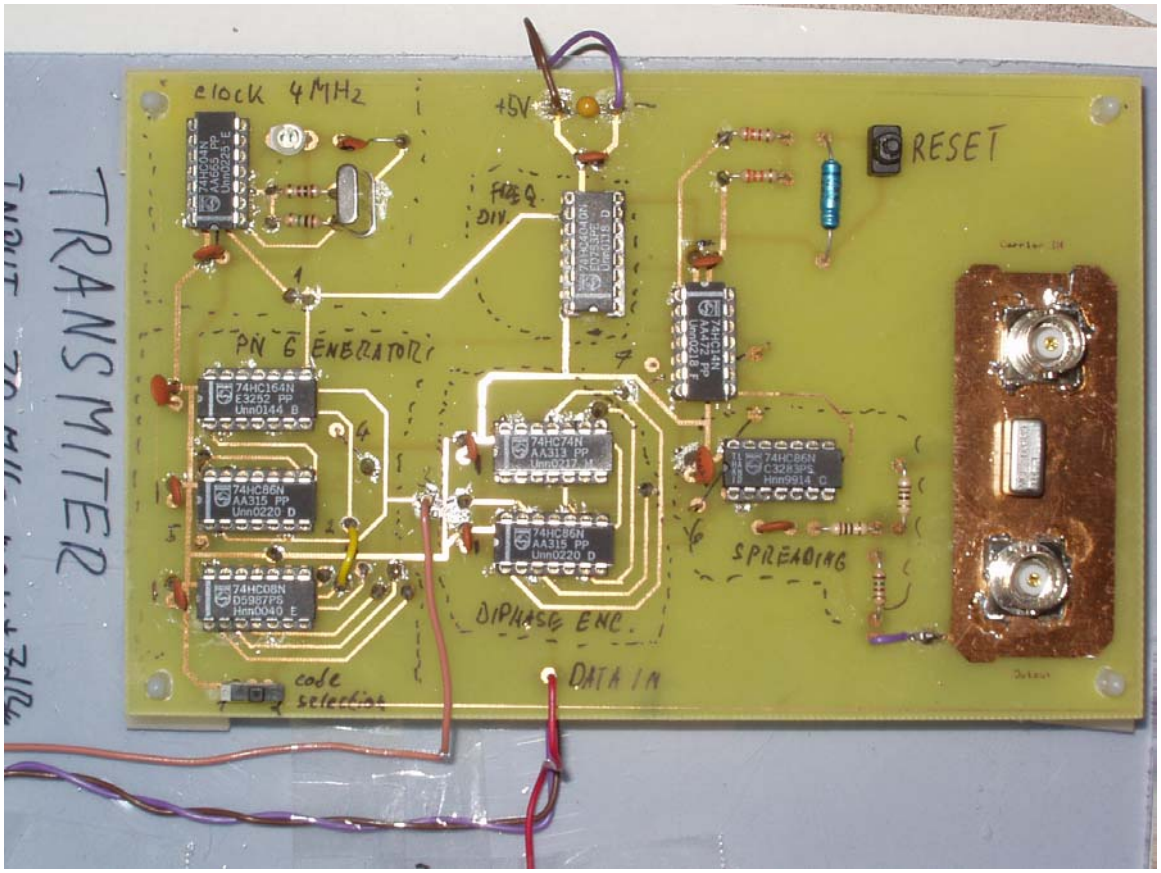


Figure 3: Receiver module

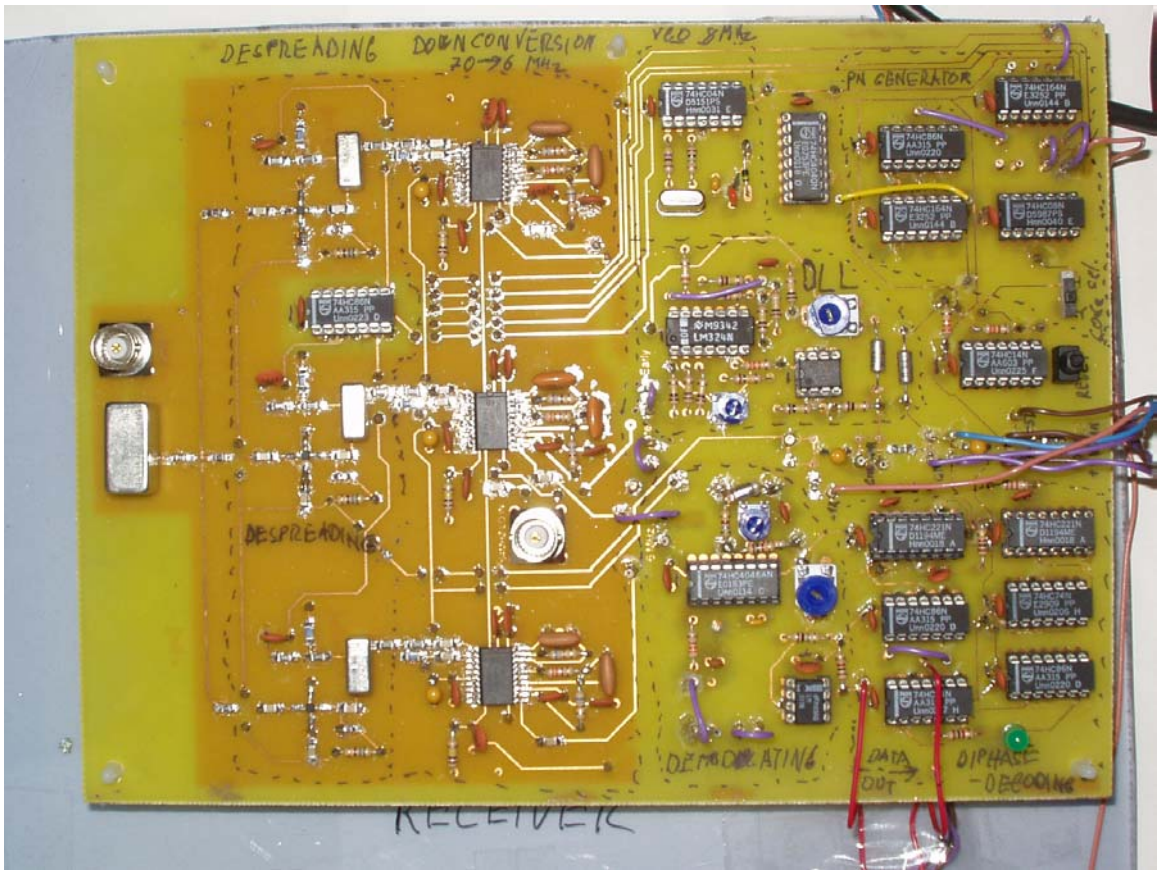
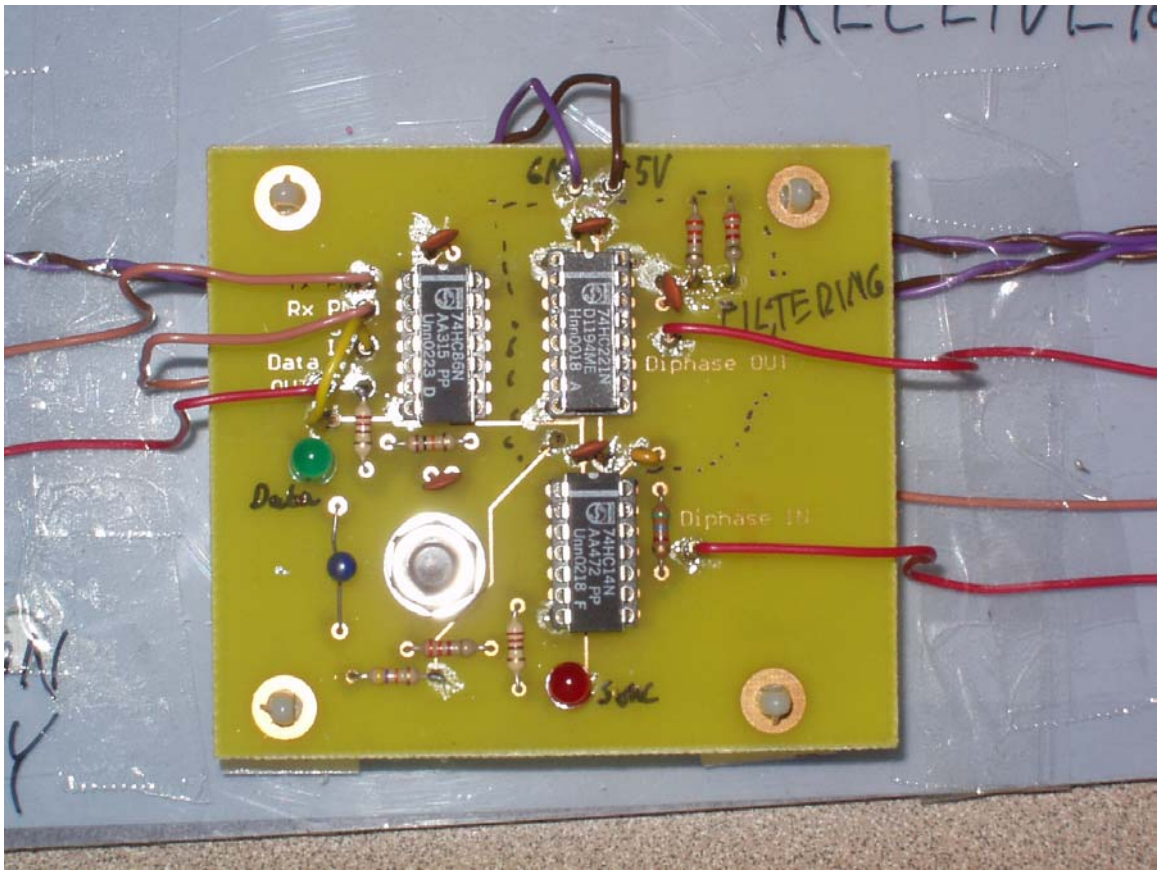


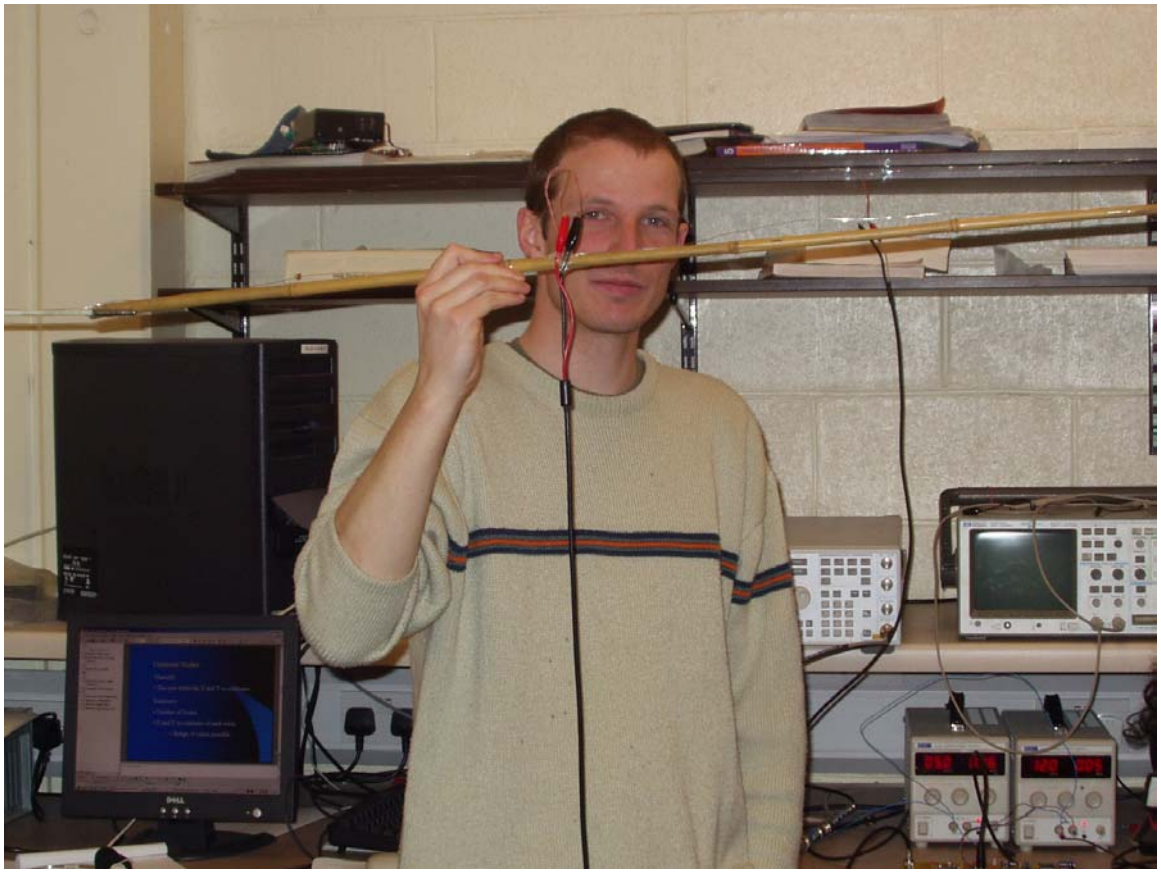
Figure 4: Demonstration circuit



- Original consideration was to connect transmitter and receiver by just coaxial cable with adequate attenuation. However when the functionality was confirmed I cannot resist the temptation to try free space transition. I constructed primitive half wavelength dipole antennas. Basically just striped 2.14 meters of wire connected to boards by coaxial cable with crocodile connectors at the other end. The result does not look very professional (see Figure 5), but it did the job. The maximum distance I achieved was to get data through the air across the lab approximately 15 meters distance and through two walls. Unfortunately CIT building is quite unsuitable for transmission testing due its metallic construction.

.

Figure 5: Dipole antennas used, one mounted by tape to shelf behind me and the other one in my hand on bamboos stick so I san move it around



3.1 Some numbers about the system

- 7 bit shift register used for pseudo noise generation => 2^7-1 = 127 bit long PN sequence
- Process gain $G_p = 10 \cdot \log_{10}(\frac{BW_{RF}}{R_{\inf o}}) = 10 \cdot \log_{10}(\frac{8 \cdot 10^6}{31.25 \cdot 10^3}) = 24dB$ where BW_{RF} is RF bandwidth taken by transmission, in this case for DSCS spectra the first lobe, which is double the PN rate. R_{info} is data rate.
- Data rate 31.25Kb/s internally, but there is no circuitry to clock data in and out.
- Maximum input carrier +7dBm is given by maximum power for transmitter output mixer specified by manufacturer. For higher levels, more powerful mixer would be needed or addition power amplifier

- Receiver sensitivity achieved –50dBm with the maximum number of attenuators I had (48dB attenuation, 6dB conversion loss in mixer, +4dBm carrier from signal generator). With this configuration synchronization was rapid (usually less than 2 seconds) and BER was in reasonable region indicated by BER figure underneath. It is possible to lower the carrier level from the generator to approximately –15dBm (using the same 48dB attenuator) when the system fails to synchronize, but the data transmission is very poor or fails at all. The question is if the mixer can operate with carrier level so low. The appropriate testing would be to keep carrier at maximum +7dBm and just add attenuators, unfortunately as I indicated above 48 dB was all I had.
- BER: The measuring technique I used is very primitive and far from being accurate, but I think it is enough to give some rough idea about performance. I simply connected DC voltmeter to output of receiver and send continuous stream of ONEs or ZEROs. The idea is that if logic ONE at the output gate has voltage let's say 4.83 V and then I measure for continuous stream of ones voltage of 4.05V I can calculate the error as 1- 4.05/4.83, similar for stream of zeros. This measurement gives following numbers:
  - +4dBm carrier, 48dB attenuator: errors in zeros 1%, errors in ones 5.9% => average gives BER=$3.45*10^{-2}$
  - Free space transmission across about 5 meters distance, no object in the path, carrier +7dBm: zeros 4.2%, ones 12.1 % => average gives BER=$8.15*10^{-2}$

- Theoretical transmitted distance can be calculated using the equations for free space transmission and known attenuation A_{tested} = -48dB for which system worked in previous measurement.
  - Free space loss $L_{free} = (\frac{\lambda}{4 \cdot \pi \cdot r})^2$, r – distance, lambda – wavelength
  - Reflection loss $L_{ref} = L_{Tx} \cdot L_{Rx} = 0.965 \cdot 0.965 = 0.931$,

    $L_{Tx} = L_{Rx} = 1 - R^2 = 1 - 0.187^2 = 0.965$, $R = \left|\frac{Z_1 - Z_2}{Z_1 + Z_2}\right| = \left|\frac{73 - 50}{73 + 50}\right| = 0.187$, Z_1 is impedance of dipole antenna (73ohms) and Z_2 is impedance of output circuit (50ohms)
  - Dipole maximum gain G_{dipole} = 2.14dB in direction transversal to dipole
  - Combining all this yields $A_{tested} = L_{free} \cdot L_{ref} \cdot G_{TXdipole} \cdot G_{RXdipole}$ and we are able to calculate distance as

    $$r = \sqrt{\frac{L_{ref} \cdot G_{TXdipole} \cdot G_{RXdipole}}{A_{tested}}} \cdot \frac{4 \cdot \pi}{\lambda} = \sqrt{\frac{0.931 \cdot 10^{0.214} \cdot 10^{0.214}}{10^{-4.8}}} \cdot \frac{4 \cdot \pi}{3 \cdot 10^8 / 70 \cdot 10^6} = 1164m$$

    so theoretically system with 48dB attenuation should be able to transmit over 1164 meters distance (of course no object in first fresnel zone)

- Material cost can be roughly estimated as something over 200 euro, which comprises of 100 euro for RF components (mixers, filter, amplifiers) and another 100 for other chips, the costs of board production and components from college store (as resistors capacitors, sockets etc.) are not counted.

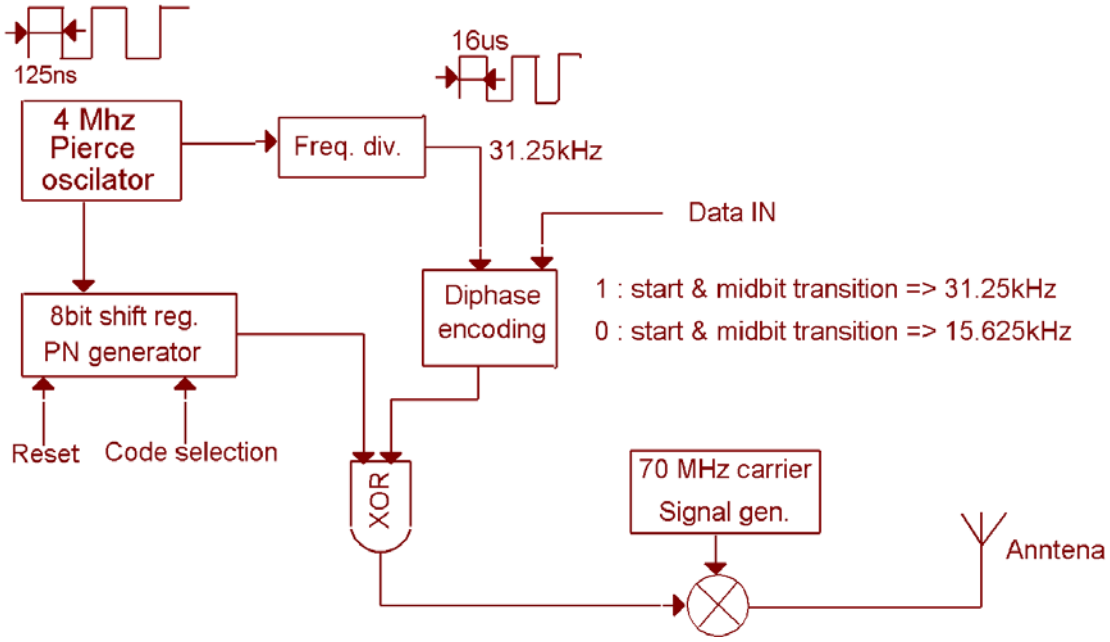
4 Design Issues

In this chapter I would like to make detailed discussion how the system work and how different components works and their implementation in system

4.1 Transmitter Brief overview

Transmitter comprises of several functional block (see Figure 6). Whole circuit is driven by 4MHz square wave Pierce oscillator. This frequency is divided in frequency divider by factor 128 giving out 31.25kHz that clocks diphase encoding circuit (details about diphase later on). 4MHz drives pseudo noise (PN) generator (For PN sequence theory refer to appendix A), in this case 8 bit shift register of which only seven bits are used to generate 127 bit long PN sequence. PN generator has two additional inputs, one is reset switch, which simply reset the PN generator to start from beginning of the sequence, and the other one is switch enabling selection one of two available PN sequences. PN sequence from PN generator and data from diphase encoder goes to XOR gate where the low bit rate data are scrambled and spread by much faster PN sequence. This is fed to double balanced mixer and modulated to 70 MHz carrier. This system uses binary shift keying (BSK) which means that output sinus wave has either 0 degree shift or 180 degree shift. This type of modulation has typical double-sided carrier suppress (DSCS) spectrum (for more details about balanced mixer and modulation refer to appendix A - circuit description). After modulation in real system there should be some kind of filter to filter out side lobes, because for effective transmission we need only main lobe of output spectra, further followed by power amplifier and finally connection to antenna. As I do not intend to have free space transmission I have no filter and amplifier and the output is going to be connected via attenuator to my receiver or directly to my transmitting antenna in which case I have just broken law and should be prosecuted so I definitely will not try it ;-) .

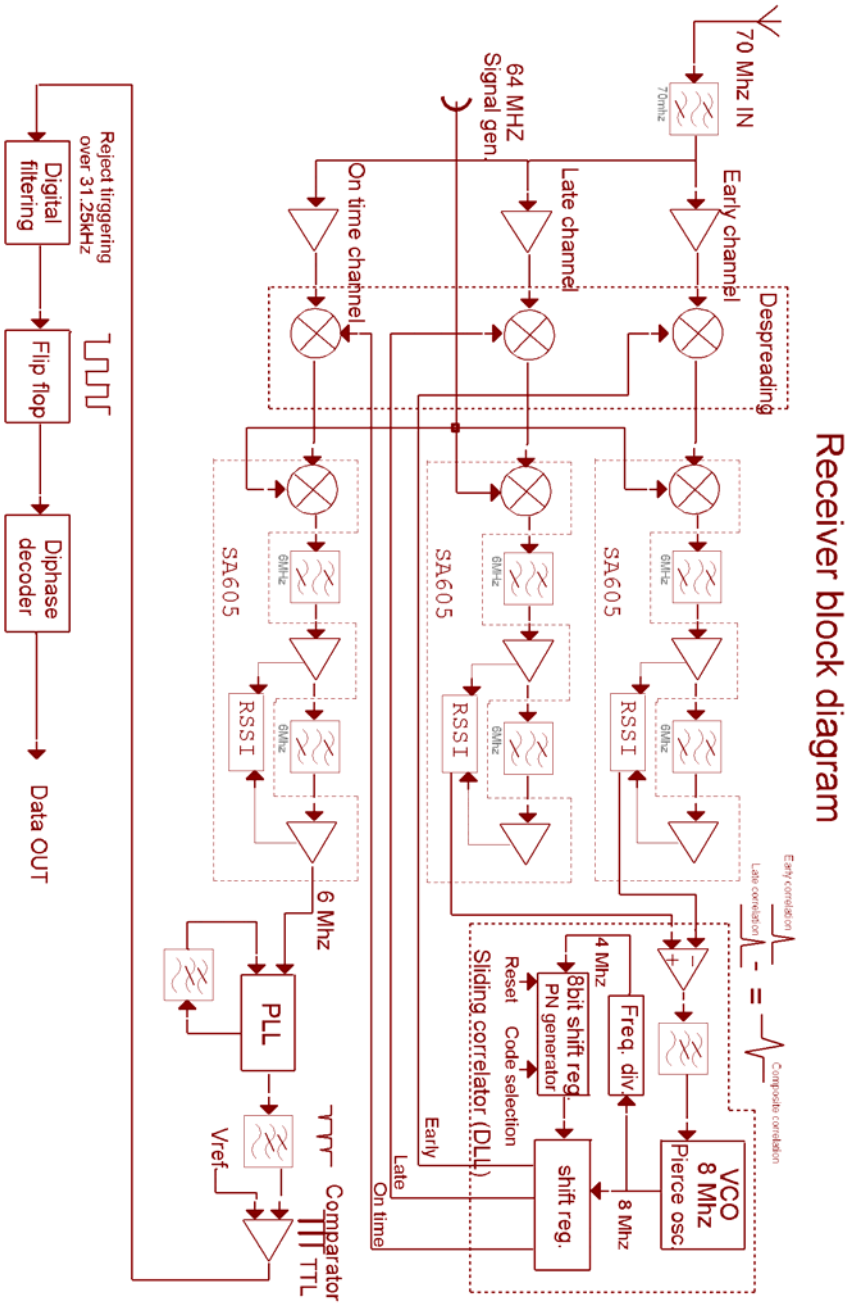
Figure 6: Transmitter block diagram



4.2 Receiver Brief overview

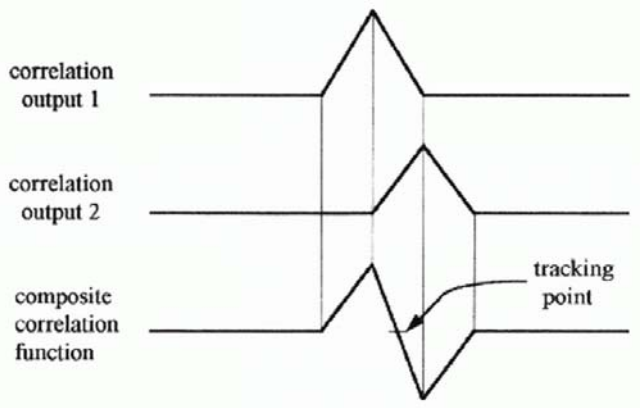
In comparison to transmitter the receiver is far more complicated. The reason for this is obvious. Transmitters job is only to scramble the data and send them to the air, but on the other hand receiver has to dig up the signal, synchronize, demodulate and decode the data that is much more complicated operation. Referring to figure 7 we can follow the signal flow through the receiver. First the signal is received by antenna or whatever input we have, then it goes through 70 MHz band pass filter and finally it is split into three identical branches. Each branch performs exactly the same operation. First there is amplifier that separates input from the rest of circuit. Next is mixer in which signal is mixed with local replica of PN code, so if the right PN sequence is applied the signal is despread and pseudo noise is gone and we are left with only data signal. Despreading is followed by integrated chip SA605 that performs several operations. First mix signal with 64MHz from local oscillator, effectively downconverting 70MHz down to 6MHz. Signal is then filtered by 6MHz band pass filter, amplified, filtered again and clipped in high gain limiter to approximately square wave at 6MHz.

Figure 7: Receiver block diagram



Chip also contains received signal strength indicator (RSSI) that monitor how good is the signal we receive. This RSSI is the major concern in first two branches (Early and Late), as the outputs of RSSI from both channels go to differencing amplifier where are subtracted then low passed and drive local voltage controlled oscillator (VCO). VCO runs at 8MHz, this is then divided by 2 to give 4MHz and drive PN generator that is exactly the same construction as in the transmitter. Local replica of PN code from PN generator goes to another shift register that is clocked by 8MHz. Three followings bits of this register are taped giving out 3 exact PN sequences shifted from each other by half chip rate (half rate of PN code => 125ns). These 3 shifted sequences are then fed back to despreading mixer in corresponding Early, Late or On time channel. This is called delay lock loop (DLL) or sliding correlator. Wonderful property of PN sequence is that autocorrelation function is constant everywhere except one point where the sequences are aligned (refer to appendix A - correlation). This fact is used by RSSI of each channel. Output of RSSI is constant all the time except one point when the sequence fed to despreading mixer is align with the sequence in transmitter. This forms correlation peak (see figure 8). As each channel is fed by slightly shifted sequence, each channel correlates at different time. First correlation occurs in Early channel, then in On time and finally in Late channel. These correlation peaks from Early and Late channel go to differencing amplifier and all the way around the DLL. The trick of DLL is that if none of the channels correlates the outputs of RSSI should be approximately constant and consequently the output of differencing amplifier should be constant as well. This means that control signal driving VCO is constant as well, so the frequency of VCO does not change. As frequency of receiver is always slightly different than in transmitter (no matter how hard we try to tune the oscillators, simply the frequency will not never match because they are not in any way connected) the PN sequence in receiver is sliding in time (sliding correlator) relatively to PN sequence in transmitter until after some finite time, depending on the difference between transmitter and receiver frequency, they will be for short period of time align, which will form correlation peak on RSSI. As I indicated above, because of shifted PN sequences fed to different channels correlation occurs in slightly shifted period of time on each channel. As this is fed to differencing amplifier this will form composite correlation function (refer to figure 8) and generate control signal to speed up or slow down the VCO. If the loop is tuned properly it will manage to stay in lock in the transition between Early and Late correlation where the composite correlation function is zero. When the loop manages to stay locked in this point, then the frequencies in transmitter and receiver are the same (in fact the receiver VCO is twice the transmitter, but exactly !!) and moreover the PN sequences in transmitter and receiver are aligned.

Figure 8: Correlation



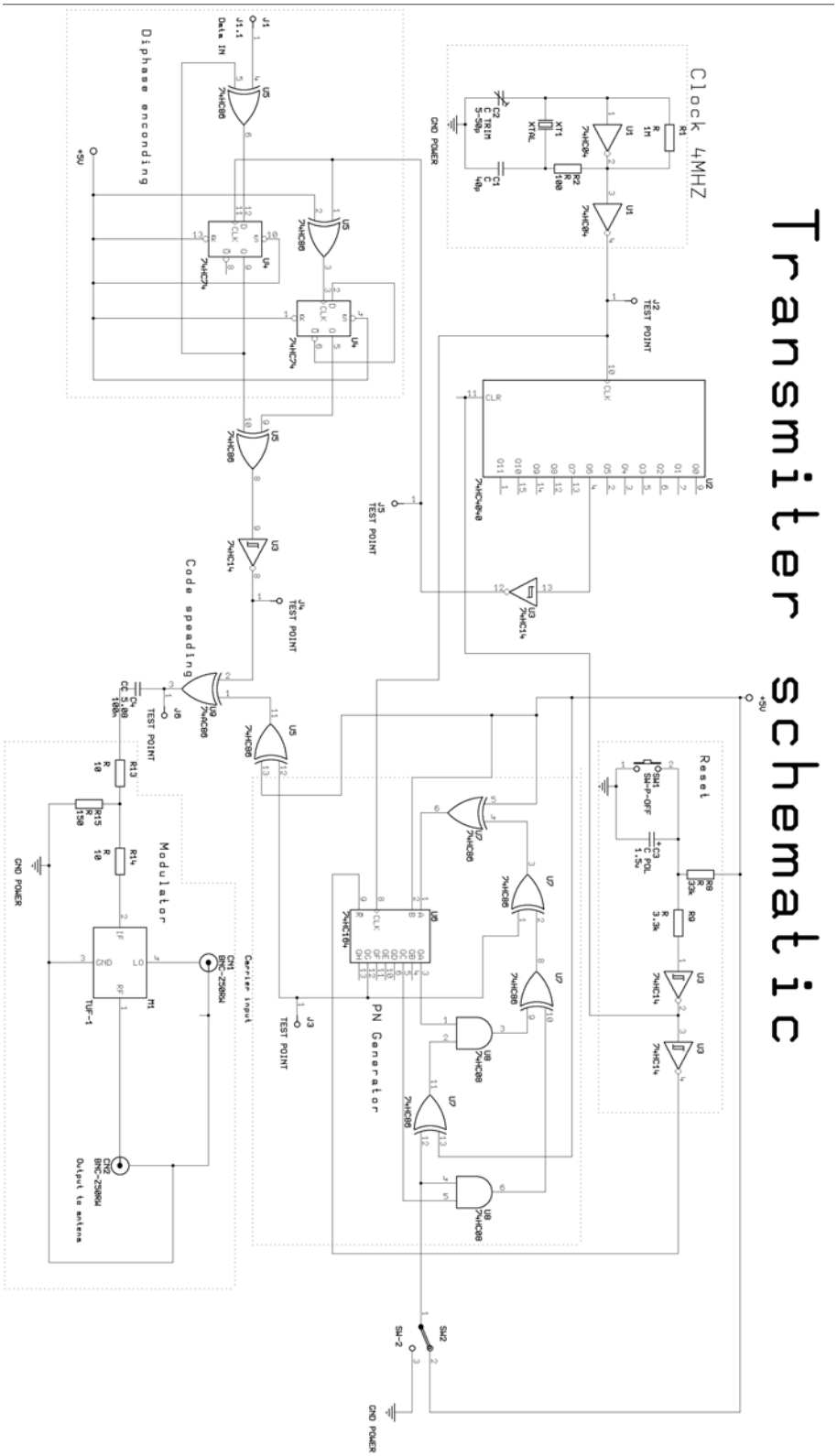
As the On time channel is fed by sequence in between the Early and Late channel, this channel should be perfectly despread and all PN noise gone. Output from On time channel (in this case no RSSI, but the output from limiter) is 6 MHz square wave with 31.25kbaud data binary phase modulated on it. Next circuit in chain is phase lock loop (PLL) that lock on 6MHz signal. It generates large error signal any time when the phase is rapidly changed, because it has to track again for lock. This rapid phase changes corresponds to our 31.25kHz data stream, so by detecting this large error signals we effectively demodulate the signal. The error signal is low pass filtered to get out only errors corresponding to major phase change (useful data). Next follows comparator circuit that converts these error peaks to compatible TTL levels. Because our transmission is not error free, sometimes the false triggering may occur. As our data rate is 31.25kbaud, there is no point in triggering faster than that, so following circuit simply

rejects any triggering faster than 31.25kHz. Next circuit converts our pulses to corresponding square wave (simple flip-flop triggered by pulse) that is fed to diphase decoder (detailed description later on), which finally regenerates the data send by transmitter.

4.3 Transmitter Detailed description

Now I would like to take a closer look at transmitter schematics.

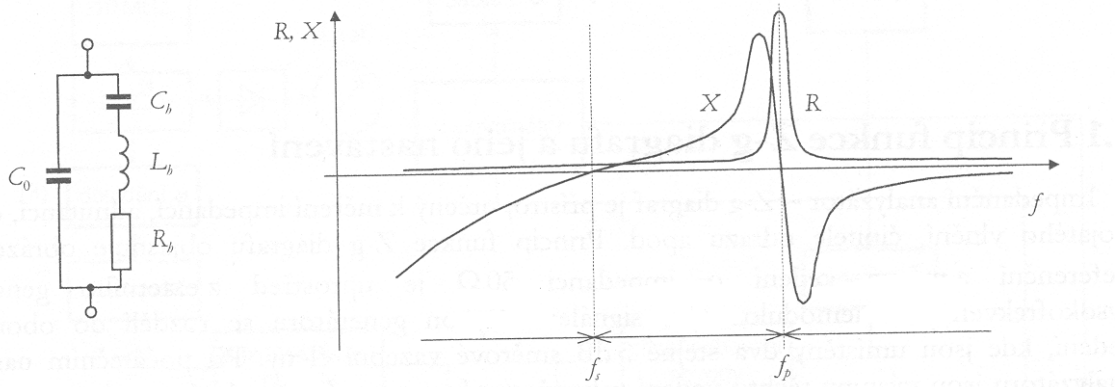
Figure 9: Transmitter schematic



4.3.1 Master clock

Original article uses different type of oscillator that has one major disadvantage. It does not work. It oscillates but the frequency is dictated by variable capacitor and not by crystal, which end up in large tuning range (2 – 8 MHz) and mainly in unstable frequency that is very undesirable for our application. I designed my own oscillator. It is 4 MHz Pierce oscillators, which can be slightly tuned by variable capacitor (tuning range here is just 2 kilohertz 3999920Hz – 4001579Hz). Inverting 74HC04 gate is used as amplifier with large gain and phase shift of 180 degree. Crystal has another 180-degree shift, so the closed loop fulfills phase condition for oscillator. Resistors R1 and R2 achieve amplitude condition. Common wiring of Pierce oscillator with CMOS gates in this configuration is that R1 should be in hundreds of kilo ohms, the exact value is not crucial (1 mega ohm is nice value), R2 should be in range of hundreds, this value should not be to large nor to small. This resistor helps to reduce the load of the crystal but if value is to high the amplitude condition is not fulfilled. The best is to build the circuit and try few values. Honestly I am not fully familiar with the function of this resistor I had to admit that my oscillator is not optimal. I think that circuit is slightly overloaded because so far I managed to burn several gates when I left system running for several hours. Value of capacitors C1 and C2 can be empirically calculated as c=150/f (capacitance is in picofarads, the frequency should be entered in megahertz). Capacitor C2 can be exchanged for variable capacitor, which enables tuning of the oscillator (see Figure 15). Crystal resonates somewhere between serial (f_s) and parallel (f_p) resonance frequency, in other words the crystal settles down somewhere between these two frequencies where both amplitude and phase condition are met (see Figure 10).

Figure 10: Equivalent circuit of the crystal & real and imaginary part of impedance with frequency

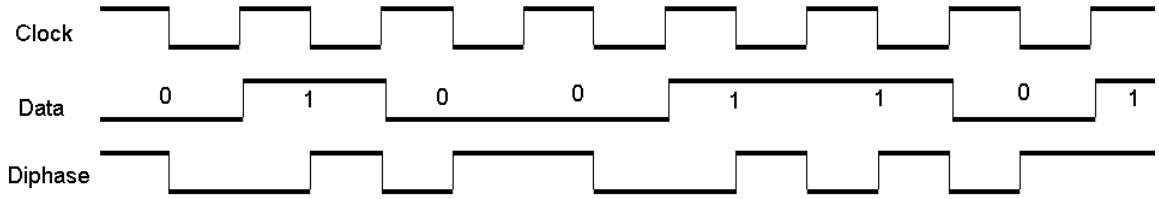


Second inverting gate is there just to shape clear square output and simply to separate oscillator from the load circuitry. Output goes to 74HC4040 binary counter which divides 4MHz by factor 128 resulting in 31.25kHz for driving diphase encoder.

4.3.2 Diphase encoder

After the signal is modulated, transmitted through the air and received by the receiver, we are not able to tell if the phase shift is 0 or 180 degree, because we have no phase reference at the receiver. The only thing we are able to detect is phase change. So in order to distinguish between ONE and ZERO, we cannot just send 0 or 180-degree shift for ONE and ZERO but we have to encode the data in some way we are able to decide if ZERO or ONE was sent. As we can in receiver detect only phase change, we decode the data in the rate of change, so ZERO will have different rate of phase change that ONE. In our case we used following scheme. ONE has start and mid bit transition and ZERO has just start bit transition (figure 11). When we apply this to continuing stream of ONEs or ZEROs we have 15.625kHz square wave corresponding to ZERO and 31.25kHz to ONE.

Figure 11: Transmitter OrCad simulation of diphase encoder (output signal is half clock slower that data)



Circuit comprises two D type flip-flop (74HC74). Upper flip-flop flips with falling edge (clock are inverted for this gate by 74HC86) of the clock that effectively means dividing frequency by factor of 2. The lower flip-flop flips with rising edge of clock and clocks in the data xored with the previous state of flip-flop. Result is constant logical ZERO or ONE (depends on previous state) for ZERO data going in and frequency division by 2 for ONE at the input (the same function like upper flip-flop). The output signals from both flip-flops are shifted by half clock rate because one is clocked by rising edge and other by falling one. This goes to XOR gate and form out the Diphase output as can be seen in figure 11.

4.3.3 PN generator

There is whole science behind generating PN sequences. This system uses the simplest way for generating PN sequence using the shift register (a little bit more can be found in appendix A – detailed circuitry). I wanted to implement two different PN codes so I need to switch between two circuit configurations. First one taps 1st and 7th bit from the shift register, XOR it and feed it back to the input, the other one taps the 3rd and 7th. Switching between these two configuration is by switch connected to logical zero or one which enables or disables one of two 74HC08 AND gates and effectively block or let pass state from 1st or 3rd bit of the register. The 3rd XOR gate in register feedback is there to avoid all zero lock up after reset of the register. It inverts first ZERO to ONE and avoids lock up. PN generator is made up of 74HC164 8bit shift register but only first 7 bits are used. According to theory, 7 bit register can generate maximal length sequence of 2^7 –1 =127 bit length.

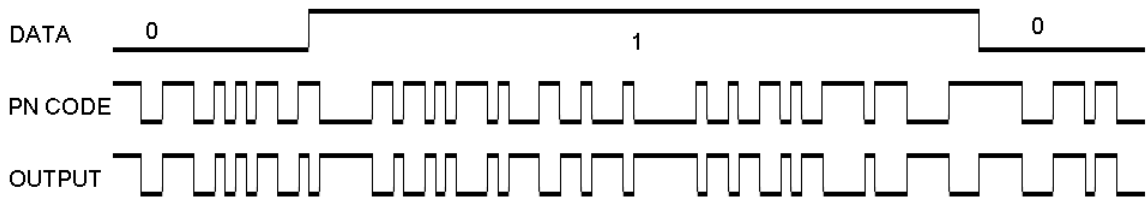
4.3.4 Reset switch

This is simple circuit to generate the pulses. It uses push button to discharge the capacitor C3 to ground. 74HC14 Schmitt triggers afterwards shape up these pulses. One output goes to frequency divider, which is reset by positive pulse, and inverted pulse goes to PN generator, which is reset by negative pulse.

4.3.5 Spreading and modulation

PN sequence and data goes to XOR gate where the data spreading occurs and data are changed to noise like signal (see figure 12).

Figure 12: Transmitter OrCad simulation of data spreading



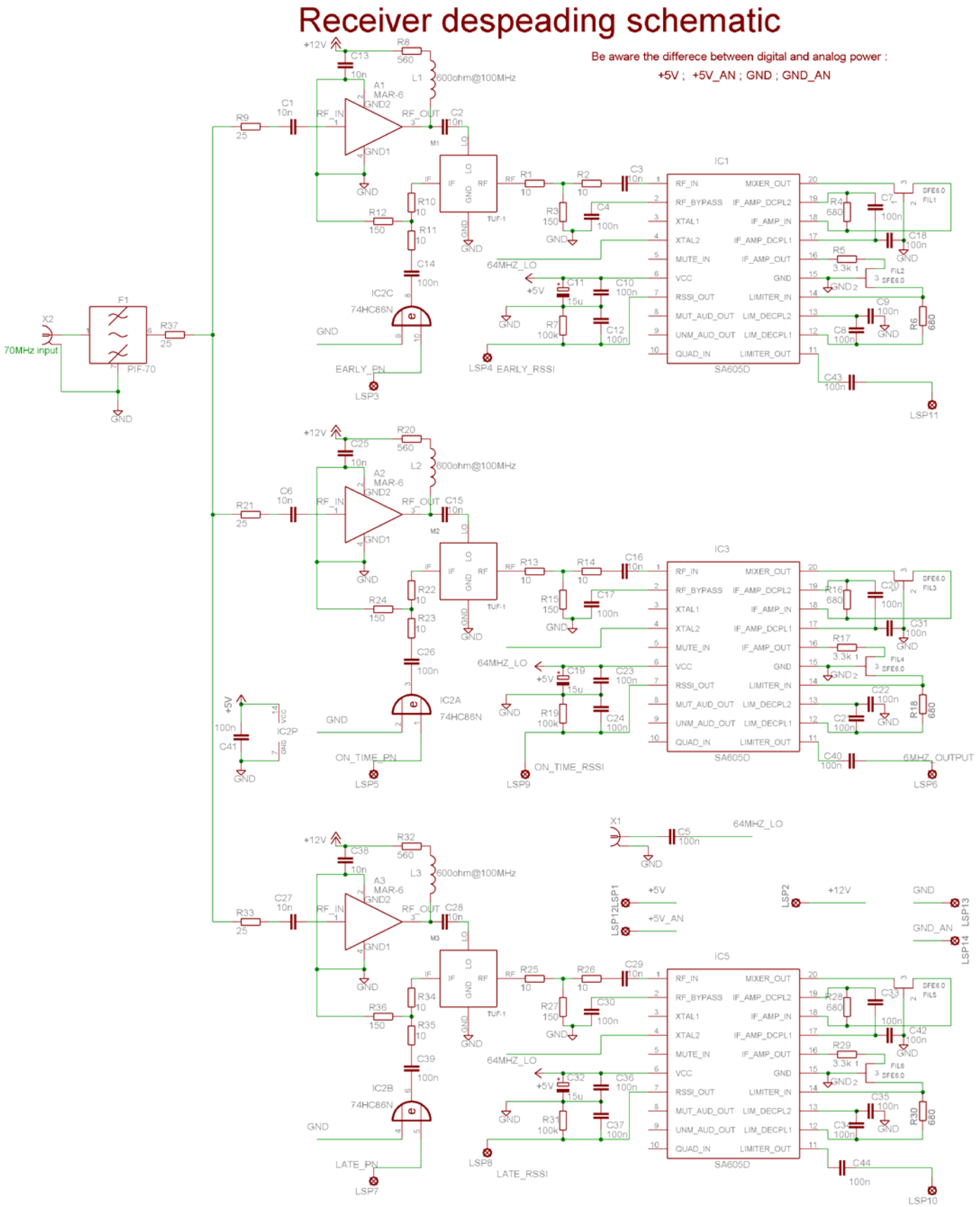
In original article 74AC86 was used for data spreading. The reason given was that AC logic has 25mA output current for both logical stages and enables to drive the mixer. Honestly I have not find any single reason why the mixer should be driven by this quite high current. Moreover when I used AC logic during the testing I managed to toast few chips (25mA can really generate bit of heat and when we multiple it by 4 gates per package it is 100mA for small central heating ;-)). So I used HC logic instead and the results are far better. The output swing voltage is about 0.4V (it was about 2.5V with AC) which corresponds to the voltage required to open two shotky diodes in mixer, the RF switching noise propagated to the output signal is also much lower than, when switching 25mA. Coupling capacitor C4 shifts the level of data from 0 to 0.4V to –0.2 0.2V. Then there is T Bridge made of 3 resistors. I really have no idea what is the purpose of this, at least it does not cause any harm. I have the impression that it should work as impedance matching, but as the input of the mixer is 50 ohms and output of HCMOS some 150 ohms it really cannot work. It can only suppress the signal reflected from mixer and vice versa. I think that system would work fine even without these resistors, but I implemented them just in case I am not right.

4.3.6 Design suggestion

As I indicated in project specification above I did not implement any circuitry for synchronizing and clocking data in the transmitter. In fact the data input in transmitter is sampled with every clock cycle of diphase encoder. For this application when I connect data input directly to constant logic ONE or ZERO it is fine as the data are sampled into circuit properly. In fact it is possible to connect faster data source as for example digital clock with frequency going up to few kilohertz, but as we approaching the data date of 31.25kbaud we can notice that the data are not sampled in the transmitter correctly. Solution for this is to implement some additional circuitry that will indicate when the input is ready to read and buffer the data before clocking into the transmitter. Ideal would be to implement some chip communicating with PC serial port so it would be easily possible to send any stream of data.

4.4 Receiver detailed description

Figure 13: Receiver despreading schematic



4.4.1 Front end

In professional receiver system there should be after the antenna some kind of low noise wide band amplifier followed by helical filter and further amplifier. Because my system is far from being professional I do not have any of these. My input goes firstly through 70MHz band pass filter PIF-70. Honestly I doubt it has any effect on the signal, because 1dB pass band of this filter is pretty wide (58-82 MHz) and reasonable stop bands where loss > 10dB is less than 16MHz and more than 280MHz. Reason for this low performance values is that the filter has constant impedance over the whole frequency range. Definitely this is not good filter for front-end application, as we ideally need only 8MHz wide band of first lobe of DSSC spectrum. The only reason why I implemented this filter is because I ordered it with other components and afterwards I felt sorry not to use it at all. I do not think it has much positive effect on the system functionality, but definitely it does do not any harm, as loss of filter is less than 1 dB.
Signal now needs to be split in three identical branches. In original schematic there are used 25 ohm resistors. Again their functionality remains quite mystery for me. I have the impression that they should work as impedance matching between the filter and MAR-6 amplifier. Output of the filter is 50 ohm, input of the amplifier is 50 ohm. The impedance which filter sees is three times parallel combination of 25 ohm resistor and 50 ohm amplifier (75 ohms), which gives 25 ohm, then there is another 25 ohm resistor at filter output, so together filter sees lovely 50 ohms. The same scenario is for what each of the amplifiers sees on its input, 3x parallel 75 ohm plus 25 ohm resistor in its own branch. This figures looks perfect theoretically, but I doubt the resistors have any effect as they are mounted on the board in different distances and board traces have some resistance and I have no idea how proper RF layout looks, so the result can be just anything. Anyway 70MHz is not so crucial frequency so I can get away with few minor (maybe even major) design errors.

4.4.2 Despreading

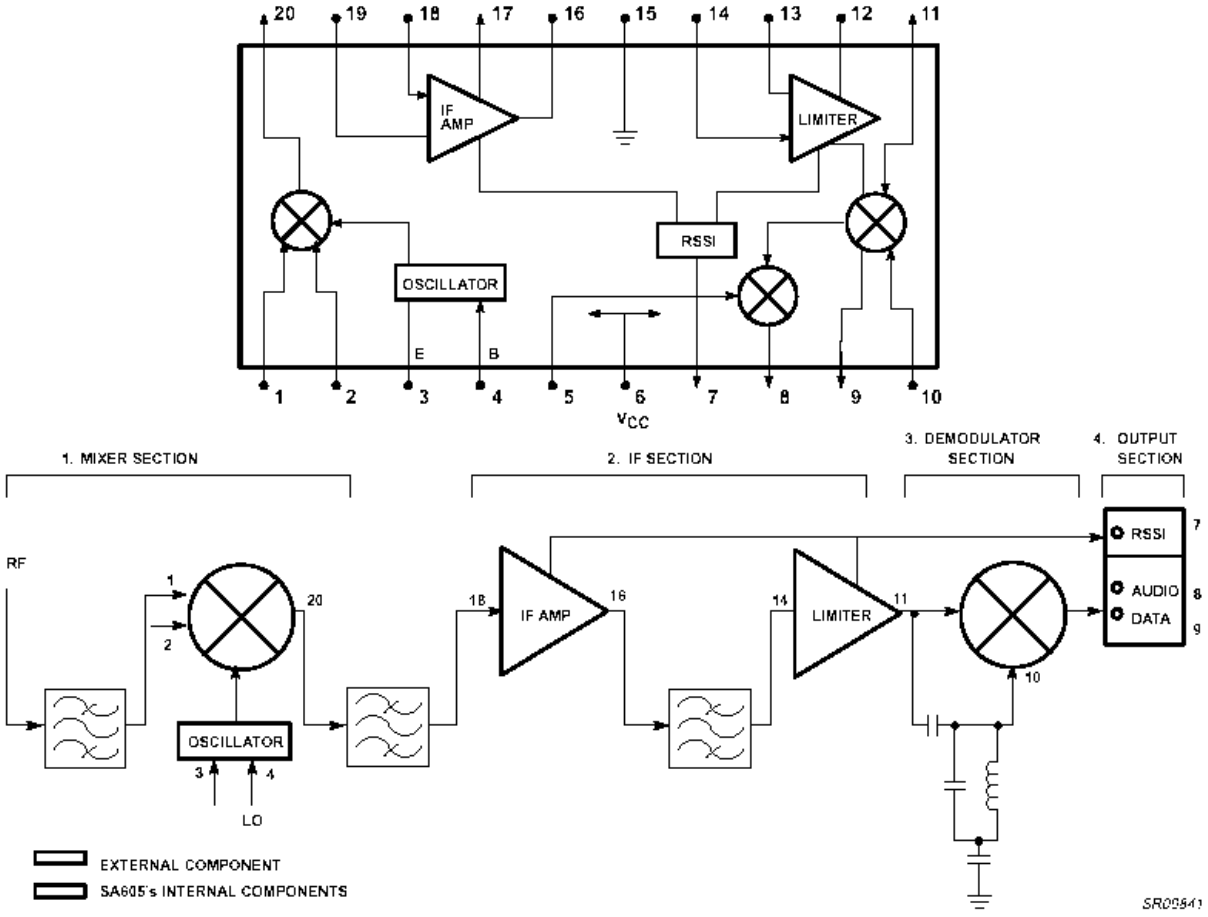
Each channel has identical component so description of only one should be satisfactory. After splitting, signal goes through coupling capacitor into MAR-6 amplifier. Usage of capacitor is recommended by amplifier data sheet, value is not crucial as long as impedance is low at signal frequency (70MHz). Again I am not fully aware about function of these MARs. Signal which I am applying at the input is usually strong enough so theoretically there should be no need for amplifier, on the other hand, amplifier can buffer the signal and separate front end from the rest of the circuitry so that any possible reflection is attenuated and not propagated back to the input. When I was measuring circuit I did not notice any additional gain to signal after passing these amplifiers, so I think that only purpose is buffering. Amplifier bios circuit is done according to datasheet using resistor, inductor and decoupling capacitor. Resistor value is calculated easily by knowing that the current should be 16mA and voltage 3.5V. When using 12V power supply resistor value is (12-3.5)/16mA = 531, nearest value of resistor is 560 ohms. Inductor – RF choke should prevent our signal in propagation to power supply, according to the datasheet the impedance of RF choke should be at least 500 ohms (10x the load impedance). Because it is the inductor, it has virtually no resistance for DC so DC power to amplifier is not affected. DC power decoupling capacitor is optional and value is not critical as long as it is big enough. Output of amplifier goes through DC blocking – decoupling capacitor to double balanced mixer TUF-1. This mixer works as despreader. Basically to reverse the operation performed in transmitter when spreading data. When correct sequence (the one synchronized with transmitter) is applied to mixer IF input the phase of the signal is changed in the way that any phase changes in signal corresponding to PN sequence are removed and only phase changes corresponding to transmitted data are left, leaving 70MHz binary phase keyed signal with 31.25kbaud data on it. IF input of the mixer is fed by same configuration as in transmitter, so I skip explanation of this part. RF output of mixer goes through mysterious resistor network to SA605 integrated chip. Function of this resistor network remains unclear for me, as it does not work as impedance matching between

mixer and SA605 (refer to SA605 application note for input impedance of SA605), it can only helps to attenuate possible reflection from the input of SA605.

4.4.3 Function of SA605

(All components reference is to the upper (Early) branch of schematic on Figure 13)
I have to admit that it cost me a lot of effort to discover full functionally of this integrated chip. I will try to cover all I have learned about this chip and maybe help somebody after me to save a lot of time digging the information elsewhere.
Original name was NE605, but now it is produced as SA605. There is also less performance version with the same functionality SA615. Chip comprises of input mixer with oscillator, IF amplifier and limiter, RSSI and demodulator mixers (see Figure 14).

Figure 14: SA605 structure



The signal first goes to the input mixer. Mixer can have balanced and unbalanced input depending how we connect pins 1 and 2 (details and example in application note). The balanced connection should be better but more complicated. This configuration uses unbalanced input. As I indicated above input circuitry is far from being impedance matched to the mixer. It has 50ohm of mixer plus 10ohms of resistor in series => 60ohms parallel to 150 resistor plus another 10 serial resistor giving 52ohms. The conversion gain of this active mixer according to datasheet should be around 12dB, the problem is that for this performance the impedance has to be matched (there is good example in application note). For 50ohm input (our case) in fact the mixer has loss of 1.2dB, so performance is greatly reduced, but I think it really does not matter in my system. Local oscillator is in fact just transistor that is ready to connect external components (like crystal) to form oscillator. We cannot use this feature because we need identical function in all 3 branches so we

cannot let each chip create its own mixing frequency, but we have to connect pin 4 to one source of frequency (in our case signal generator). According to application note, to achieve mixing, the level of this frequency should be somewhere between 200 – 600 mV RMS. During the testing I was using 250 mV RMS all the time, but I found that system works fine even when we go as low as 40mV.

Output of the mixer will contain product of mixing 70 and 64 MHz, which is 6MHz. This has to be filtered in narrow band pass filter (3dB range about 130kHz wide) and fed to IF section amplifier (about 40dB gain). The filters used SFE6.0 are designed for other application than we use and consequently have different parameters than the chip has been designed for. There should be proper impedance matching to match output of the mixer to the impedance of filter and then input of the amplifier. Chip assumes impedance of filter around 1.5kohm, but this filter has impedance only 470 (see datasheet). But again this system is not performance oriented ☺. Usage of R4 680 ohm resistor remains mystery. I tried both application with and without this resistor and it makes no visible difference. I suspect this resistor that it affect biasing condition of amplifier, reduces gain of amplifier and help to stabilize the system (more further on). There is crucial bug in original schematic with biasing capacitors, configuration of C7, C8, C9, C18 has to be as it is in my schematic otherwise the amplifiers will not work!

After passing through first amplifier, signal goes through another SFE6.0 and is fed to high gain limiter. R6 680 resistor has probably the same function as in previous amplifier. R5 resistor is there is introduce desired mid-stage loss. Chip is designed to expect 12 dB loss between amplifiers for optimal function of RSSI. There is detailed discussion about this loss in application note including nice example. Again there is missing impedance matching.

Limiter should amplify the signal so much that the sinus wave will be clipped to square wave and effectively will get rid of any garbage on sinus wave. After this, chip contains additional section for FM demodulation, which is not used in this application. The output we use is RSSI output in Early and Late channels and Limiter_OUT in On time channel. RSSI indicates the strength of the signal we receive and is used to indicate correlation as described above in brief overview. It is current to voltage converter and for proper operation require R7 of value 100k, capacitor C12 is optional. As mentioned above, chip is designed to assume 12dB loss between amplifiers, this results in optimal and linear indication of RSSI. It should have sensitivity 0.5V for every 20dB of change in input signal. For our application it means that, if we have process gain of 24dB, the level of this indicator should vary in range slightly above 0.5V for the case when the sequence is synchronized and when not. In fact during the testing this change was about 0.6V so perfectly corresponding to theory.

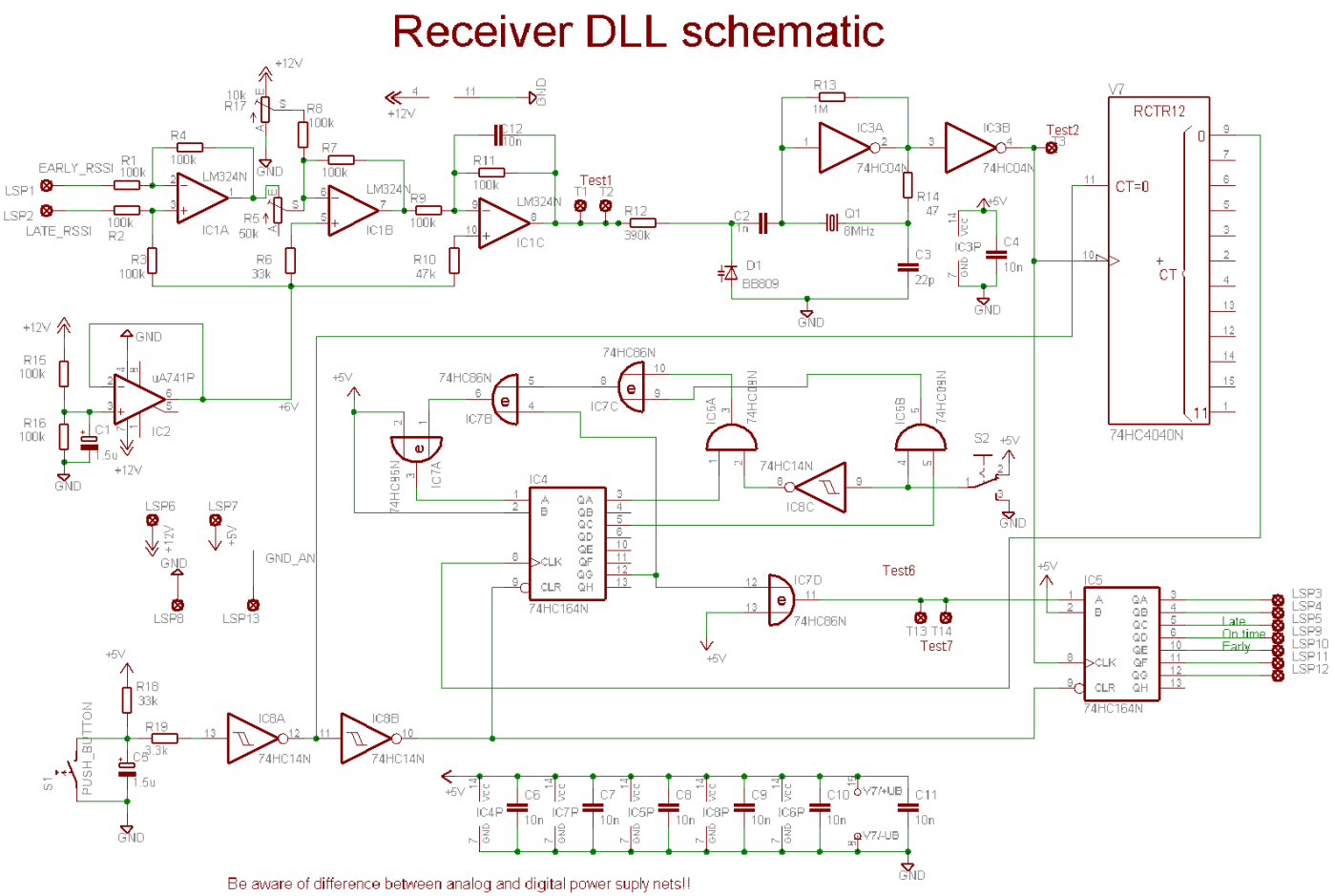
As I mentioned above we have to be careful about chip stability. Because both amplifiers together give about 100 dB of gain there is real danger that signal from output is coupled (radiated through air etc.) to input and amplified in a loop. This is concern when we have noisy layout without proper suppression, so our case. Application note says that we should not proceed with the design if there is output from RSSI more that 250 mV with no signal at the input. The advice is to introduce more interstage loss between the amplifiers and to reduce amplifiers gain. Interstage loss is determined by R5 and filter loss. I used 3.3kohm resistor to have interstage loss about 12dB, in original design the resistor is bigger value and the loss is about 18 dB. During testing I have tried various values of resistor but it does not have much of effect so I left the recommended value of 12 dB. The second thing is to reduce the gain of the amplifiers. I do not know much about this, only as I mentioned above I think that 680 ohm resistors limits the current floating to amplifiers and consequently reduces gain. No matter how hard I was trying to experiment with these values I did not achieved recommended value less than 250 mV with no signal at the input. In fact I managed to achieve this figure, but only with the SA605 itself with all the other circuitry switched off. As

soon as we switch the rest of the circuitry on, the digital switching noise is so terrible that the output from RSSI goes up to some 2V with no signal at the input. I do not know what conclusion should I make from this, but as long as the system works and RSSI indicate correctly synchronization I just leave it as it is.
The last thing to mention about SA605 is the powering issue. The decoupling capacitors are really essential. The configuration has to be as recommended by application note. It means it needs one big tantalum capacitor for stable power supply and in parallel one smaller ceramic with quite high self-resonant frequency to filter out higher frequency noise. Ideally this should be probably SMD.

4.4.4 DLL

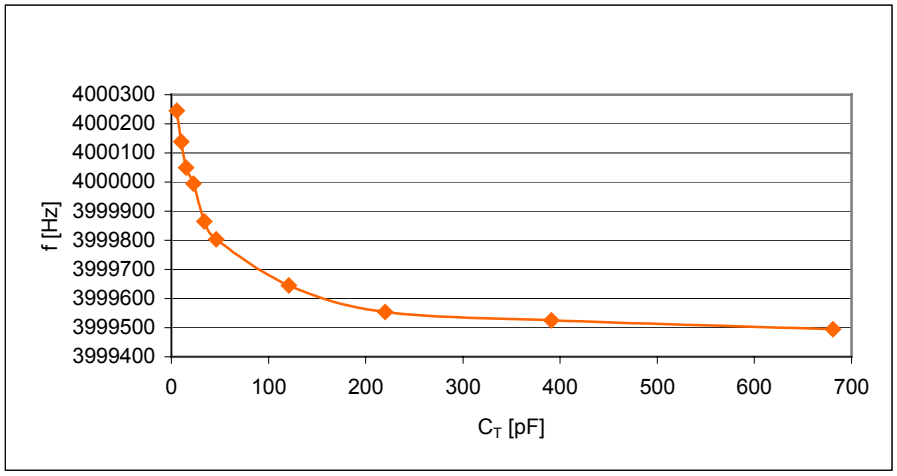
The RSSI outputs from Early and Late channels are tapped and go to DLL amplifier section. This section is made up of one LM324N chip containing 4 opams in one package. First amplifier is using as differencing amplifier with unity gain. I used component values from original schematic, but I am not sure they are appropriate because as I said above RSSI is current to voltage converter, which needs 100k resistor to ground at its output. The problem is when we connect diff. amp. with 100k resistors at input and another 100k in feedback, so effectively 200k parallel to that previous 100k at RSSI output we probably seriously affect RSSI output. I guess that the input impedance of diff. amp. should be much higher but again as in many previous cases as long as the circuit is working I am happy and I will touch anything to avoid failure. Next stage is just inverter with variable gain. In original schematic this was just unity gain inverter, but I discovered that it is not enough for dynamic range of VCO, so I introduced 50kohm variable resistor that enable to change gain in loop in range 2 to approximately 100. The other function of this stage is also to add some current through variable resistor R17 and effectively shift the level of output signal. This way we can set up center frequency of VCO. Next amplifier is used as active low pass filter at about 150Hz to reject any fast and noise like responses in loop.

Figure 15: Receiver DLL schematic



There is also another uA741 opamp that is used simply for 6V reference needed for differencing amplifier, as 6V is in the middle of tuning range of VCO.
Next stage of DLL is VCO. In original schematic, the type of oscillator used was completely different. I decided to use again Pierce oscillator, exactly the same configuration as in transmitter. The reasons for that were following. The original oscillator was rather complicated and containing a lot of components comparing to Pierce configuration and after all the output was converted to TTL square wave so there is no difference in output between Pierce and this one. The other problem was with varicap diode. Varicap diodes become nearly extinct species, so I was quite lucky to get any varicap at all. BB212 used in original schematic is discontinued product for several years and there is no replacement. I found out that all of semiconductors manufactures nearly stopped production of varicap diodes without any replacement. The reason for this is for me unknown and I really wonder what modern electronic uses instead of varicapes. Nevertheless I managed to get few BB809 varicap diodes in one shop so I have to come up with implementation using these. It is true that nearly any diode can be used as varicap, but why to worry about that when we can get component tailored for this purpose.
Taking in consideration that different type of oscillator with different type of varicap is used changes slightly the requirements for loop design. The original BB212 has capacity up to 500pF in 12V tuning range, the BB809 that I used has only 50pF tuning range (see datasheets of varicaps). This probably results in different gain needed in the loop and in need to add variable gain in second stage of amplifier. On the other hand 5 - 50 pF range of capacity of the varicap is very positive value for Pierce oscillator as the values of capacitors should be around 20pf (empiric formula 150/8MHz as discussed in transmitter section). As we see, the tuning of the varicap is exactly in the optimal region for this oscillator. I tried also to implement another 5-50pF capacitance trimmer parallel to varicap for better tuning the center frequency of VCO, but the capacitance was to high and the oscillator got to the low slope region and consequently the tuning range of oscillator was not large enough (see Figure 16). Value of R12 should be as large as possible. Originally I used just 100k and the system works fine, but then I tried 390k and I discovered that noise in analog circuit was greatly improved as less signal from oscillator can propagate back to analog circuitry through larger resistor. C2 value was changed randomly and exact value does not matter as long as it is big enough. It is just DC block separating oscillator and DC voltage for tuning varicap. Value of R14 was picked up experimentally, to best fit the tuning range of VCO to frequency of transmitter. When the value was higher the oscillator center frequency was higher than twice the transmitter so it was impossible to achieve lock. In the other words we need the VCO to be tunable exactly around twice the frequency of transmitter. Frequency of transmitter was set to 4001000Hz so the receiver needs to be tunable in range of few hundreds hertz around 8002000Hz. Honestly it is quite tricky business to achieve match like this. The crystals used have tolerance 50ppm that yields 400 Hz tolerance for 8MHz and 200 Hz for 4MHz crystal, so we see that in worst case scenario we have shifted frequency by 600Hz just by crystal tolerances not counting the other factors. So it seems that tuning of components value is necessary for each individual piece of crystal.

Figure 16: Frequency dependance of 4MHz Pierce oscilator on tunnig capacitance C_T

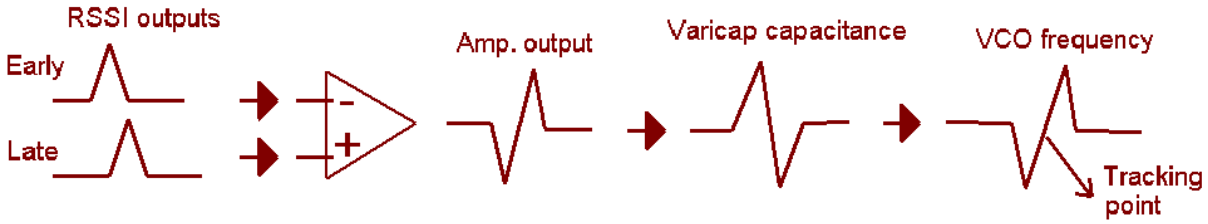


Output from the oscillator is divided by factor of 2 and drives PN generator that is exactly the same as in transmitter. PN sequence from PN generator goes to another 74HC164 shift register, which is clocked by 8MHz. Three following bits of register are tapped forming Late, On time and Late sequence. As the clock of register is twice the rate of PN code the sequence shift between following bits in register is half this, so 125ns. This is fed back to despreading mixers.

Now I try to describe function of DLL, it is bit tricky and causes small headache, but with bit of imagination and paper and pen to draw a picture it is possible to understand. I try to describe the case when frequency of receiver clock is smaller than two times frequency of transmitter (see Figure 17) :

1) $f_t > 2f_r$
2) Early channel correlates first forming positive peak on RSSI output
3) After passing through all 3 stages of inverting amplifiers (3x minus gives minus) we have negative peak at low pass filter output
4) Low voltage peak means increasing capacitance of varicap
5) Bigger capacitance decrease frequency of VCO
6) Difference between transmitter and receiver clock frequency gets bigger
7) Sliding of transmitter and receiver PN sequences is faster
8) Early channel passes its correlation peak and output of Early RSSI is getting lower, Late channel starts to correlate.
9) Negative peak at output of differencing amplifier gets smaller and smaller and continues to positive voltage peak as Late correlation takes over
10) High voltage peak means decreasing capacitance of varicap
11) Lower capacitance -> higher frequency of VCO
12) The frequency of receiver is increasing until it gets bigger than twice the frequency of transmitter (in fact it is loop overshoot)
13) Sliding of PN sequence is stopped and then reversed when Late correlation takes over.
14) As sliding direction is reversed Late channel cease to correlate and correlation returns back to Early channel
15) By this the lock is achieved and system settles itself down in this point in between Early and Late correlation

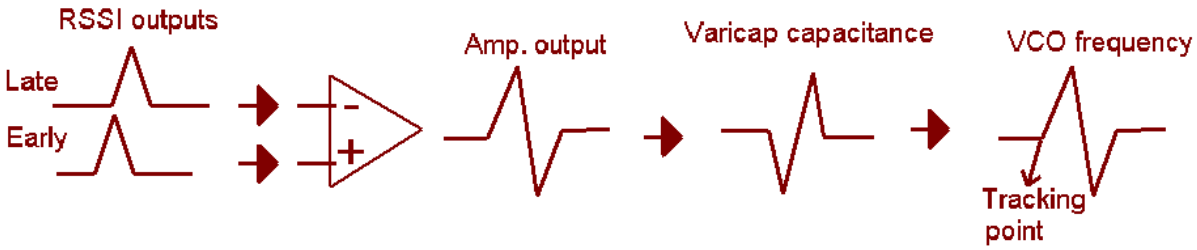
Figure 17: DLL synchronization process



Similar discussion can be done for the case when frequency of receiver is bigger than twice the transmitter clock.

It is quite interesting to mention that the system manages to lock even when the inputs to differencing amplifier are swapped (Early and Late RSSI), but this version only works when frequency of receiver is bigger than transmitter. The system is then locked in different point, effectively pushing the correlation in front, so the On time channel needs to be fed by different bit from shift register to achieve correct despreading (Figure 18).

Figure 18: Reversed DLL synchronization process



4.4.5 On time channel demodulation

Assuming we achieved lock in our DLL and managed to get rid of PN noise we are left only with our transmitted data that now needs to be demodulated. As I mentioned in description of SA605, signal is down-converted to 6MHz square wave. This 6MHz going out from Limiter_OUT of On time channel is far from being perfect square wave, it is more like bit crippled and clipped sinus wave with peak to peak deviation of approximately 700mV that is enough to fed PLL (though coupling capacitor C15). I just would like to mention that demodulation part is completely redesigned from original circuit so I am 100% to blame for any bugs and imperfection in this part.

Figure 19: Receiver demodulation & diphase decoding schematic

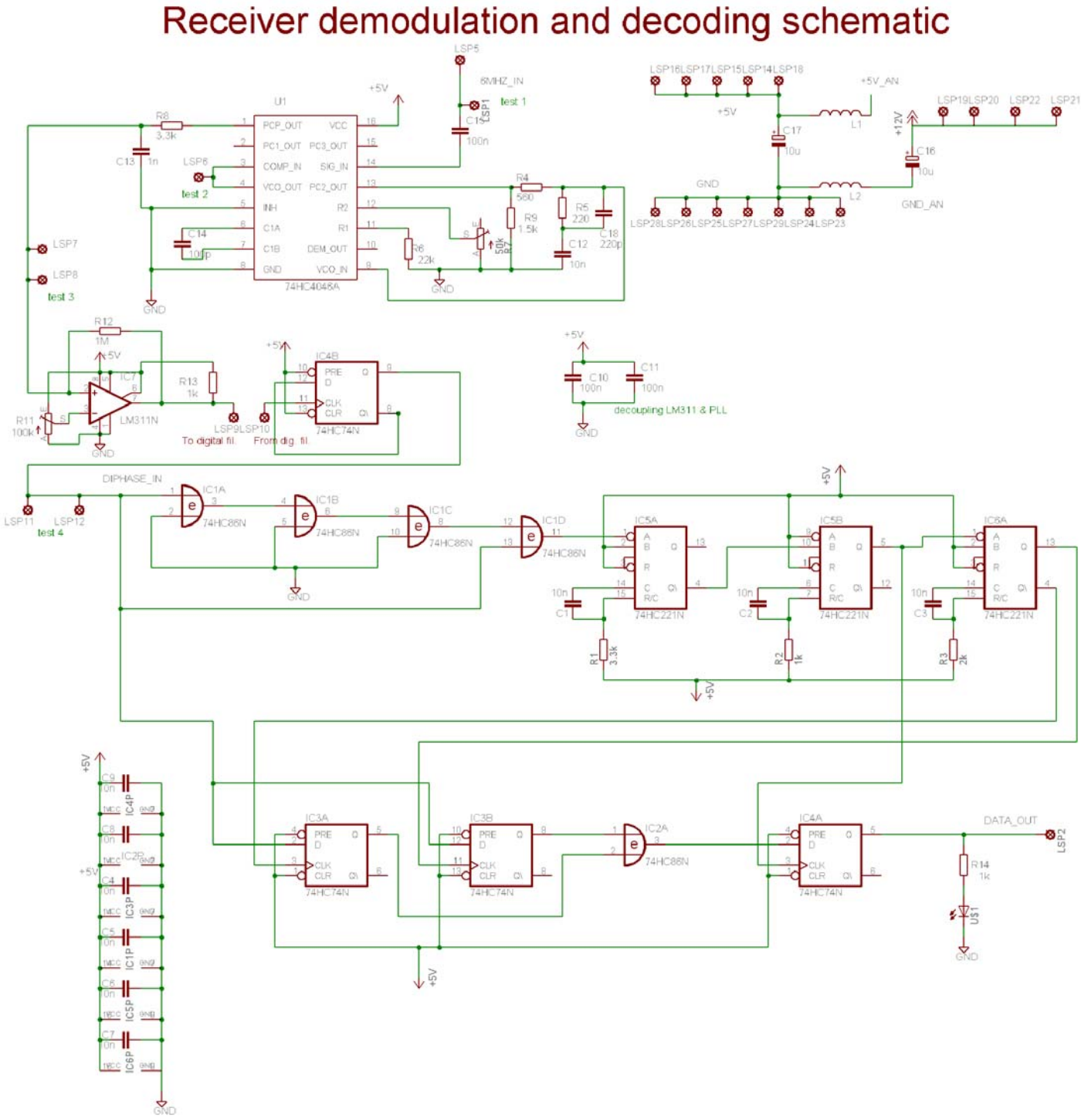
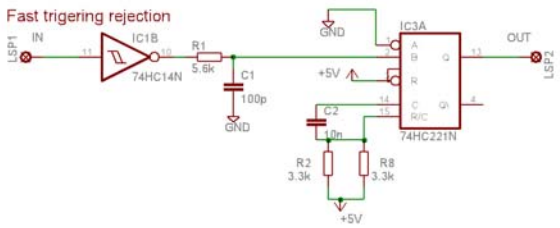


Figure 20: Receiver fast trigger rejection schematic



4.4.5.1 PLL

PLL is a typical problem of control engineering and there are plenty of books full of theory about PLLs. The amount of information available is sometimes so overwhelming that it is quite hard to find any easy and ready to apply description. The PLL used is common cheap Philips 74HC4046A. This loop should be able to work up to 17MHz so our application for tracking 6MHz is far in safe region. PLL consists of VCO and three different phase comparators. I recommend to have a look on datasheet as it is quite user friendly and contains valuable information and examples. PLL lock frequency (2Fl) is set by one external resistor (on pin R1) and capacitor (on pins C1A C1B), another external resistor (pin R2) sets up frequency offset. Then there has to be external low pass filter between phase comparator output and input driving VCO. In my application I used phase comparator 2 as the duty factor for this comparator is irrelevant (in contrary to comparator 1) and it has TTL level output PCP_OUT (missing in comparator 3). My idea was that PLL should lock on 6MHz going from SA605 and any time there is a rapid phase change (corresponding to my data) the loop should generate large error signal as it has to hunt for 6MHz again. Philips provides lovely old-fashioned DOS program specially made for designing external component for this chip. Unfortunately this program caused me more trouble than help. I used it for calculating the external component for my loop and surprisingly when I afterwards build the circuit it did not work. So I had no other option than to dig a bit into PLL design theory and calculate component values by hand. I really do not know what to think about the program because its results are usually in hard contrast to results I get calculating by hand, i.e. when I design component for 5% overshoot and then enter these value to program it tells me that overshoots are 90% and system is unstable etc. I revised figures I was using several times but with no luck. I build the circuit according to my calculation and tested it using signal generator with the same modulated signal as going from SA605 and it worked fine. The more I was surprised when I connected PLL to rest of circuit and it simply did not lock at all. So I rushed back to calculation revised my figures several more times with no result. Afterward I was desperately experimenting for two days with components values when I finally managed PLL to lock. Paradoxlly the components finally used in low pass filter are slight modification of result from the design program. Another irony is that according to the design program resistor R9 should bring small constant phase error and decrease jitter in the loop. In fact this ‘small’ error is 180 degrees, so the loop operates in completely different condition that I wanted. It is out of lock all the time generating large error signal and when rapid phase change occurs the loop locks and error signal is small.

As a output for following circuitry I did not use DEM_OUT because error level swing on this pin is quite small and goes randomly up and down depending on which of the output transistors of phase comparator is open, so it would be quite tricky job to implement any kind of comparator. PCP_OUT pin provides TTL level pulses corresponding to phase difference in phase comparator. As the output pulses corresponds to phase error of 6MHz signal it needs to be low pass filtered to get out only significant errors corresponding to received data. Filter is simple RC combination with serious attenuation above 1MHz. Original assumption was that loop will be in lock for most of the time and consequently PCP_OUT level will be HIGH for most of the time with burst of LOW level pulses corresponding to rapid phase change in signal. After low pass filtering it should be nearly 5V with low voltage peaks corresponding to phase change. It would be reasonable to set up comparator level slightly under 5V to get correct triggering. In fact as I mentioned above the loop has constant phase error of 180 degrees that results after low pass filter in steady state value of about 2.5V with low level peaks corresponding to phase change. Consequently the comparator level has to be set somewhere around 2V to get correct triggering.

4.4.5.2 Comparator

Comparator is quite fast single voltage LM311 comparator. It is used in typical application as taken from datasheet. Trimmer R11 sets up reference voltage for comparator. R12 in feedback adds small hystersis of several milivolts. I think it is worth to mention that when I was playing with comparator levels between 4-5volts, as it would be needed if my PLL works correctly I discovered interesting fact omitted in datasheet. Comparator cannot use reference voltage closer than approximately 1V to power supply i.e. when I wanted comparator level of 4.5V when using 5V power supply it did not work. The highest effective level is about 4,1V. The same case is for low voltage. This is probably due some voltage drops on transistors inside the chip.

4.4.5.3 Fast trigger rejection and flip-flop

After our signal is compared in comparator we should get nice lovely TTL pulses corresponding to our received data. The reality is not so nice. In fact there is much more pulses. Usually more than one trigger occurs corresponding to phase change, this is caused by that the error signal crosses comparator level several times and there are very short glitch like triggers. I also noticed that comparator triggering itself couples back to input and causes this glitch triggers. As we expect our data coming not faster that with 31.25kHz rate, we can simply get rid of these additional fast glitches. First the signal goes to 74HC14 Schmitt trigger to get clear TTL levels HIGH or LOW, because comparator output sometimes contains pulses that are not properly finished to LOW level. Next we let this go through simple RC low pass filter with serious attenuation somewhere around 1MHz. This merges glitch like triggers to one pulse. Following circuit is 74HC221 monostable. This circuit generates pulses of length dictated by its RC timing component and it will not trigger another pulse until previous is finished. Timing component there are selected to generate pulses of about 12us long so any triggers occurring faster than this are simply rejected. After we filtered fast triggers we can convert pulses to what we had in transmitter – square wave, 31.25kHz corresponding to ONE and 15.625kHz corresponding to ZERO. This is done in simple 74HC74 flip-flop. It is clocked by incoming pulses and flips with every pulse coming, forming square wave.

4.4.6 Diphase decoder

We feed square wave coming from flip flop to input of diphase decoder. There is quite detailed description of function of this circuit in original article so I can skip another piece of boring text. Just to mention, RC timing components in original schematic are completely wrong, values which I use works fine. There is one additional flip-flop added at the end of diphase decoder. The data appearing on XOR gate IC2A are valid only in certain time interval, the rest of the time the output is wrong. In original schematic there was delta demodulator which was clocking in the data in correct time interval, as I do not have anything like this I had to add this additional flip-flop to read data in correct time intervals and converts them to pure ONEs and ZEROs. After this chip I added LED diode to indicate signal level at data output and to have at least something bit eye-catching on my dull board ☺.

4.4.7 Design suggestion

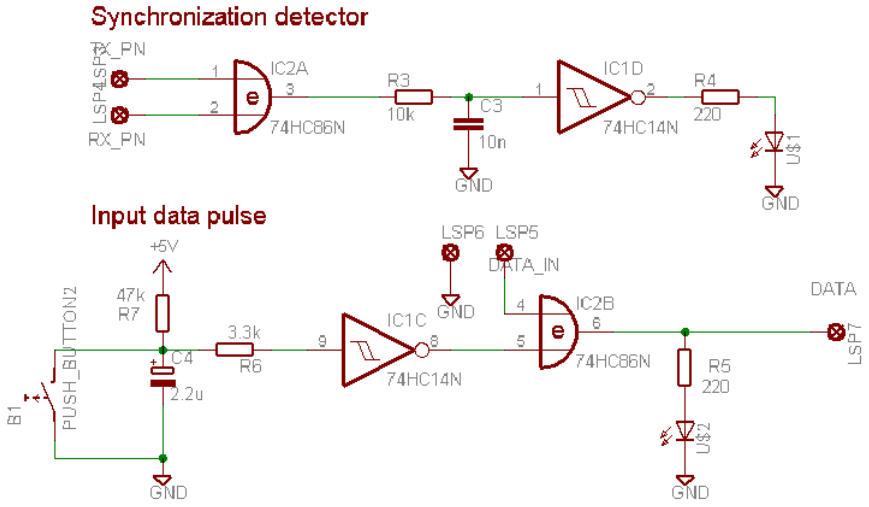
In receiver module there are a lot of issues that could be improved. Firstly, great improvement would be proper impedance matching around SA605 chip. I mean to implement balanced impedance matched input to the mixer and proper impedance matching with SFE6.0 . Next nice to have thing is implementation of local oscillator on the board, so there would be no need for external signal generator. This could be achieved for example by some faster type of PLL. Another

improvement could be probably done in DLL. As this is closed loop system it is challenging problem for control engineering. The loop parameters should be properly calculated, to achieve optimal overshoots, settling time, and jitter in the loop. Another problem is input of differencing amplifier. I am nearly sure (according to my testing and measurement) that it does a lot of harm to output of RSSI. PLL section should be completely revised, as I am not very satisfied that it operates only by accident. Last issue is the same as in the transmitter. It would be nice to have some additional circuitry to extract received data, ideally to PC.

4.5 Demonstration circuitry

In order to demonstrate in some way the results of my project and to show PN sequence synchronization and data transmission I just feel it would be appropriate to have something more than just few wires connecting to +5V or ground. So I implemented bit of additional simple circuitry for demonstration purposes.

Figure 21: Demonstration circuit schematic



4.5.1 Lock detector

Lock detector is simple circuit that indicates if transmitter and receiver PN sequences are synchronized. It consists of XOR gate which one input is connected to transmitter PN sequence and the other to receiver PN sequence. When sequences are locked they are exactly inverted, this is because the pins I used for taping sequences from transmitter and receiver. So when system is locked the output of XOR gate is ZERO with minor glitches caused by synchronization jitter. To get rid of this jitter we use low pass RC filter followed by Schmitt trigger. Synchronization is indicated by off state of diode.

4.5.2 Input data pulse

This is simple structure used also in reset buttons to generate pulses, components values were used without any calculation by pure guess, and so pulse is bit long. XOR gate is there if we want to bypass pulse button and apply any other source of data (like digital clock), if not used this pin has to be connected to GND provided.

4.6 RF issues & noise

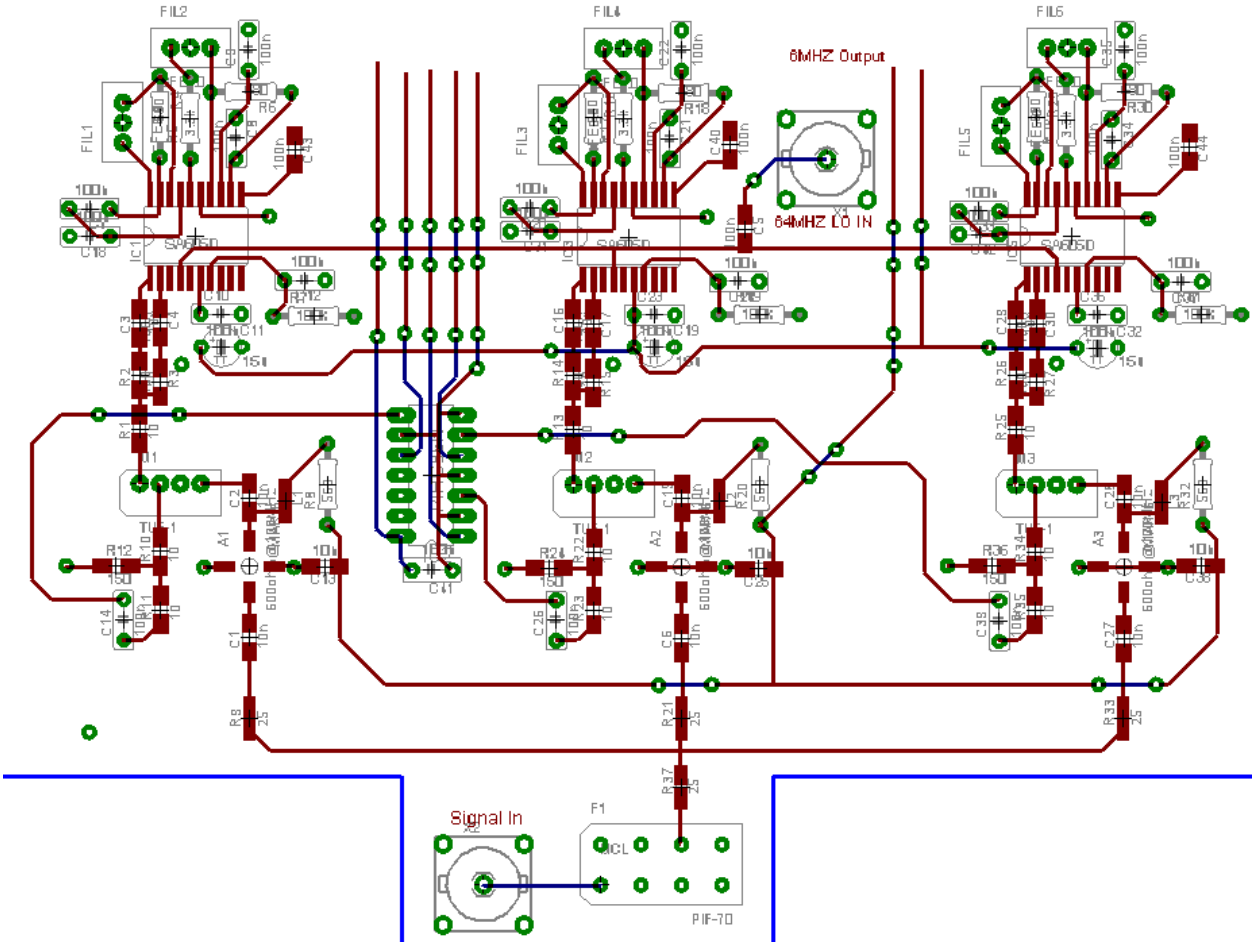
This system is only 70MHz, but even the 70MHz is RF frequency so proper RF layout should be applied (words taken from data sheet ☺). The problem is that I have no experience with RF layout and no matter how hard I tried I did not find any book or other information source describing what proper RF layout is (I really got impression that to obtain this knowledge is whole-life quest and only brave and honorable knight is permitted to gain it).

4.6.1 RF layout

There are only a few parts in my system operating at 70MHz. In transmitter it is just two BNC connectors and one mixer. On receiver side, it is front end of the system followed by three branches of Early, On time and Late channels up to inputs of SA605s. As the wavelength of 70MHz is 4,28 meters, the rule of thumb gives safe track length up to 21 cm (up to 1/20 of wavelength we do not need to worry about track impedance). I tried to keep my track safely under this boundary. The next thing I feel would be appropriate for RF layout is proper ground plane. Unfortunately I have no further knowledge how this ground plane should be implemented, how large should it be, from which side of board, how far from the components, etc. Nevertheless I tried to have ground plane under my RF components in transmitter and receiver. Double-sided board unfortunately does not give much option for trace routing so the plane is not always so compact as I wished. Another issue is that I found in book in information sources that 1.6mm thickness of standard FR-4 epoxy used is too large for any effective shielding property of ground plane. With RF circuits we also have to care about radiation from track, like various shapes of tracks radiates more or less effectively. Probably the right angle bent of the track is not very good idea, on the other hand all vias and component mounts are in fact right angle bended track and maybe even worse, so there is not much we can do about that.

Probably only problem of RF layout we can affect in my primitive system is how to effectively split the power of the signal in three identical channels of receiver. I came with idea that I do not need three identical channels, but only two. I need only Early and Late channels to be symmetrical, as I am using RSSI from these channels. On time channel balance to the other two does not matter because the RSSI is not used there. It is simply fine as long as I am getting some signal to this channel for further demodulation. To achieve symmetry for these two I used simple crossroad with 90degree angles, signal from filter going from down, left and right identical branches of Early and Late channels and On time channel going up (check the artwork on Figure 22). After splitting, I tried Early and Late channel RF tracks and component location to be exactly the same sizes and distances to maintain symmetry. On time channel is also identical, only track is bit shorter (check the artwork on Figure 22). The only asymmetry is in distribution of 64MHz local oscillator signal, I tried to make this asymmetry very small and place connection to BNC connector as close to the center of the board as was possible. I do not know if all this effort has any effect, but at least I could say I have done the best I can.

Figure 22: Receiver despreading artwork



Next thing that needs to be highly considered when dealing with higher frequencies is the components packages. 70MHz should directly indicate that we have to go for surface mount technology. Just to mention few justifications for this. Standard disk plate through hole capacitor have self-resonant frequency in best case 50MHz for small capacitance values and really short leads (for typical application of 10nF capacitor it is about 30MHz), it means that at 70MHz this capacitor will have quite significant impedance that is very undesirable effect. Basically any through hole components adds few nanohenries of inductance caused by its leads, which forms lovely resonating circuit at higher frequencies and degrade the component. There were several unsuccessful predecessors in my project and not considering another crucial errors they made one of them was using through hole components which probably are the edge of functionality at this frequency and the other one went even far more and used breadboard and then was surprised that parasitic capacitance killed whole signal. Due the component delivery problems I did not used SMDs all around the RF part, but only in essential part on the way of 70MHz signal.

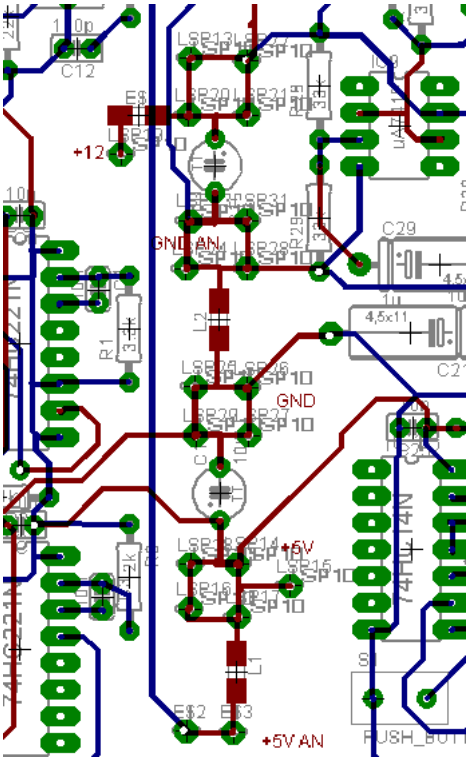
4.6.2 Noise suppression

When I was constructing first pieces of my project I thought that I will not worry much about the noise, because I do not expect any good performance from my system and I did not implement much noise suppression. Surprisingly first test and measurement shown that even in my system I will have to worry about noise as it highly affect functionality of the system. Digital circuits are real nightmare. When I looked on spectrum analyzer my 4 and 8MHz switching signals generated unbelievable garbage up to several hundreds megahertz. Explanation for this is quite simple, digital MOS circuit has wonderful property that they do not drag virtually any current as long as

they remain in steady state. But when clocking edge occurs real hell starts. All components switch at the same moment, it means in MOS technology in charging and discharging all transistors. So for very short period of rising edge the circuit drags serious current. It sounds really unbelievable but system like mine can drag several tens of amperes for period when rising edge persists. The current can be calculated by knowing time of rising edge, capacitance load of the gates and number of switching gates (refer to book in information sources). This high current floating through power supply nets generates massive magnetic fields so effectively our noise. Other problem is that power supply and nets cannot support such large current flow, that ends up in power supply voltage bounce. This bounce affects mainly analog circuit that requires precise voltage reference. Transferring this problem to our system, the voltage bounce caused by digital switching is something as high as 300mV, if we take in consideration that for example RSSI indicator output has swing of 600mV it is evident that it is going to cause us a lot of trouble.

To fight this I tried to implement any countermeasures I was able to think about. You can see in schematic and in artwork that each chip has its own decoupling capacitor as close to the chip as possible to reduce current flowing in power nets. Next common technique is to have large capacitors at power inputs into the board and around sensitive circuitry (as SA605). Unfortunately this is not enough as the noise is spread through power nets like plague. We have to clearly distinguish between analog and digital circuitry and have separate power supplies for each of them. One technique to achieve this is to separate power nets by noise suppressors (some kind of inductor for high impedance at higher frequencies). I have no experience with this, so probably the components I used are not optimal but it helps to improve things a bit (see Figure 23). Major improvement was in transmitter, when after applying this ferrite the noise propagated to output signal was reduced drastically. See the schematics and artworks for the implementation of inductors. Good reference for noise suppression tips is book in information sources.

Figure 23: Ferrites suppressors between digital and analog power supplies & and big tantalum capacitor at input of power supply



4.7 Comments to original article

After spending few months with playing with this system I feel justified to have few comments to the article my project is based on. It is very nice and instructive when one wants to get familiar with principles of spread spectrum direct sequence. I found it very helpful in getting theoretical knowledge around spread spectrum and for this purpose it can be strongly recommended. The other part of article is not so bright, honestly I wonder how somebody who build that system can publish so incomplete documentation full of bugs. This results in that the reproduction of the system is nearly as hard as to start from the scratch. I hope that I will do better and if somebody will try to use my report for system reproduction it would be more helpful.

5 Construction & testing

Even when there was mentioned a lot of construction issues in design chapter I think that there is more to say about practical issues concerning physical realization of the project. Originally I wanted to support my conclusions by measured results from oscilloscope and spectrum analyzer, unfortunately the equipment in the lab was not ready to connect to PC and I did not have more time to set up the equipment.

5.1 Board construction

After considering RF issues I decided that standard type of board is enough and that there is no need for special low loss material. I took advantage of that college has facility to produce standard FR-4 epoxy boards. All what is needed is to print artwork on transparency and exposure, develop and etch the board. For printing artwork I always used printer best quality and 1200dpi resolution. Each page has to be printed twice and copies has to be put one over another when exposuring for getting optimal results. Procedure I used was that I printed all four pages for double-sided board (2 copies for each side), two identical copies put together using tape and then put together both sides of the board. Board then can be inserted in between these transparencies put into exposure. First we exposure one side then turn around to the other. Follows the developing and etching process. System was constructed in several steps. First I confirmed functionality of separate functional digital blocks and constructed it on the breadboard. Second stage was to construct system on PCB. I used multiple PCBs for different functional block (Transmitter, Receiver despreading, Receiver DLL, Receiver demodulation, Receiver diphase decoder). In this stage I had all I need for functional system, problem was that it never work properly when connected together. Interconnecting wires was picking and radiating so much noise that system tends to work in one moment and fails in the other when I slightly moved the wires. It was necessary to merge everything on one board for proper functionality. Final version is one board for transmitter (surprisingly the same one I used in stage 2 ☺), another board for receiver and finally one small board for demonstration purposes. Receiver board is not very nice, because dimension of board was bigger than etching tank so had had to turn board several times in tank to etch all areas. Consequently there are over-etched areas on the board and another areas with redundant copper. To be perfectly honest the small board for demonstration purposes does not contain only circuit for demonstration but also fast trigger rejection circuit, which I forgot to implement on receiver board and then it, was to late to start over again.

To be able to manipulate whole system I mounted all three boards on transparent prospect. This also helped to reduce the ratnest that I had in power and other wiring and mount the wires neatly on the prospect. Result is quite compact and it is possible to move the system around without affecting functionality.

5.2 Radiation & crosstalk

As I discussed above the noise cause really a lot of trouble. Just to mention few notes concerning free space transmission. When I connected receiving antenna to spectrum analyzer, it was nearly impossible to distinguish transmitted signal from noise, only first lobe of the spectrum was visible few dBs above the noise level. The problem is that this high noise level is caused by circuit itself. When I switched all digital circuits off noise level went dramatically down by some 30 dB, this only supports the fact that my layout and design is very poor and it would be appropriate to implement some additional shielding.

Another drawback of my design I noticed was that I did not care about unused gates in packages and left them unconnected. The noise radiated through air and propagated in power nets is distributed on this unused gates and force them to trigger, so effectively I have more switching gates generating more noise that is necessary. In proper design unused gates should be connected to corresponding logical signal to avoid switching.

I also noticed pretty high crosstalk between some tracks. Especially long parallel track feeding dispreading mixers by PN sequences have crosstalk nearly about 0,5V. I was aware of this danger when I was routing the board, but there was not much other options to route the tracks.

5.3 Surface mount technology

I have to admit that when I first discovered that I will have to use SMDs I was quite horrified, because SMD is treated as something mysterious belonging only to area of precise industrial robots. The truth is exactly in contrary. As long as we have reasonable large packages (I used 1206 where it possible) it is actually much simpler that through hole component. For standard through hole component we have to drill hole, worry if we managed to align bottom and top side of the board, then bend and cut the leads and finally fight with solder to stick to copper. There are no such problems with SMDs. We simply cover the pad with solder, then use the pump to smoothen and remove redundant solder on the pad, finally, using tweezers, lay the components on the pad and heat the pins with the iron tip for a while and component is soldered! Easy, convenient and elegant. If I knew this from beginning I would probably design board to use much more SMDs, as the soldering is much faster.

5.4 Tunable components in system

Ideally system should not have any tunable components. This scenario fits the best industrial production of thousands of boards of same construction, where additional tuning would be very costly. As this is amateur prototype we are quite happy to have something to play with and few tunable components are necessity.

Transmitter has just one tunable component. It is variable capacitor in master clock that enables to slightly change the frequency of transmitter clock.

In receiver we are more plentiful. There is variable resistor R17 in DLL, which enable tuning of center frequency of VCO, by shifting level of second amplifier. There is another trimmer R5 nearby that controls the gain of DLL as discussed above in DLL. Another trimmer is in PLL for tuning frequency offset of center frequency of PLLs' VCO. It could be replaced by approximately 20k resistor, but the trimmer was more convenient for setting up system. Last variable component is trimmer in comparator circuit. This simply set up the level where comparator should trigger. I do

not know why, but this trimmer is quite sensitive and usually needs slight adjustment when moving the board.

It is also worth to mention about the buttons and switches. There is one push button in transmitter and one in receiver. Pressing these simply reset the corresponding circuit PN generator and frequency divider, so the sequence should start from beginning. The purpose of slide button on transmitter and receiver is selecting between two available PN sequences. We have code number ONE and code number TWO corresponding to numbers written on the board next to switches.

5.5 Setting up the system

Maybe after few years somebody will want to experiment with board that I have built. I am pretty sure that board will not start working immediately but will need proper retuning. These are the steps that I can suggest for setting up board for operation (assuming all circuits are OK and no track is broken):

1) Short-circuit the inputs of differencing amplifier (Late and Early inputs).
2) Set up the trimmer R17 to middle position and using frequency counter monitor center frequency of VCO
3) Using variable capacitor set up exactly half of this frequency in transmitter clock. If the tuning range is not enough, it is essential to change center frequency of VCO (using R17) until tunable regions of both oscillators overlap. And then try again. R17 should not be set to any boundary position, as there would be no tuning range left for DLL.
4) Connect the spectrum analyzer to transmitter output and check for correct DSCS spectrum at the output
5) Connect transmitter to receiver via few attenuators (15 dB is nice value) and using probe connected to spectrum analyzer check if signal is propagated correctly in all channels on 70MHz circuitry and through filters around SA605s.
6) Connect the DLL and check if the systems locks
7) Connect the PLL to output of On time channel and check that PLL is locking on 6MHz
8) Check if output of the loop (after low pass filter) produces errors corresponding to transmitted data. If not, check if also data, and not only PN sequence are really send by transmitter. I noticed several times that diphase decoder in transmitter locks up, so if this happens reset the circuit.
9) Set comparator level to right trigger level.
10) Pray for success ☺
11) Switch on signal generators and power supply, switch transmitter and receiver to use same codes, repeat your prayers

6 Documentation reference & information sources

In this chapter I would like to present additional material that I produced during my project and some of the useful information sources. I burned a lot of useful additional material on CD. I try to attach it to the report, but I cannot guarantee that it will remain there.

6.1 OrCad simulation

My original idea was to perform simulation of whole system before proceeding with design. Unfortunately it shoved up to be impossible, because there were no appropriate models for lot of the circuitry. However it was very helpful to simulate some parts of the project. I used OrCad to simulate nearly whole transmitter. I was not so successful with receiver. The only part I was able to simulate was differencing amplifier, for rest of the circuitry there was not any usable model in OrCad.

On CD there is in directory OrCad archived all stuff I did in OrCad, project name is SS. There is a lot unusable garbage in this project as I was playing around. I personally think that only thing that can be of any use are transmitter pages, which simulate whole transmitter including output mixer. I also included OrCad tutorial, help and library list in OrCad/Doc directory.

6.2 Board schematic & artwork

To my great surprise college does not have any official PCB design program. Originally I wanted to draw everything in OrCad but afterward I discovered that there is no license for Layout module on college and trial version is useless. Mr. Brian Donovan is in possession of 2 programs. One is some old DOS program, which I was not brave enough to try to use. The other one is pretty good EasyPC, but unfortunately installation disk was not anywhere to find. Anyway, these were his personal property, so legally, I cannot use it. Another program on college is Pads. Problem is that there is only one license protected by hardware key and it was used by postgraduate students. This problem with software results in that transmitter is designed in different program than receiver.

6.2.1 Formica

For transmitter, I used software from my home college. Unfortunately I was able to get only trial version, which has component limitation. It was enough for transmitter, as it is much smaller than the receiver. It is called Formica, unfortunately for you (Irish, English or whoever you are), it is czech software, so all documentation is only in Czech language. Nevertheless the program itself is in English so it should not be problem to use it just to view my schematic and artwork. First impression is that it is really user unfriendly, but believe me, once you gets familiar with it, it is very handy.
CD includes program installation files in Formica\program\form430w.zip. It installs two modules. First is Schematic, for schematic design, which can be used to view transmitter schematic formica\project\transmitter.sch. The other one is Layout for artwork drawing. Transmitter artwork is in formica\project\transmmiter_final.PCB. There is also help and documentation files in formica\program directory, but as I said above it is in Czech.

6.2.2 Eagle

One day I had a bit of luck and managed to download from internet cracked full version of another program. I know it is not legal and moreover it should no be used for official college project, but I had no other option.

This is German software. It is nothing amazing and it keeps crashing all the time but it did the job. CD contains program installation in Eagle\program\eagle-4.09r2e .exe and cracked license file licence.key for which you will be asked after installation. I hope that I enclosed the right one as I had collected several cracks before I found one that is working correctly. Faulty crack can be recognized by that it is unable to load previously saved file.

After installation copy the Eagle\projects directory in programs projects directory and program will recognize it as valid project called SS. It is also necessary to link my library file with the special components models I had to add to program. It is in eagle\project\lib .

Project contains:

- despreading – dispreading part of receiver
- DLL – delay locked loop of receiver
- diphase_decoder – demodulation and diphase decoding part of receiver
- receiver.brd – all sections merged together on one PCB (final version of the board), this is just artwork copied together from other sections so there is no schematic for this.
- Some other boards which I used during construction for testing and, which preceded final version of system
- Some schematic drawings I used for presentation and this report

6.3 Other material on CD

Datasheets	
74hc4046a	PLL datasheet and design program
mar-6	IF amplifier datasheet and biasing recommendations
tuf-1	balanced mixer datasheet
sa605	SA605 chip datasheet and application note
SFE60	SFE6.0 ceramic filter
Other	datasheets for other components used in system
Minicircuit	letter from minicircuits distributor from UK, with all contact details
Original circuit	schematic pictures of circuit from original article
Pictures	Photos of whole system for better visualization of the problem
Presentation	Several presentation in PowerPoint which I used for presenting my project
Theory	Few interesting theory articles
Report	electronic version of this report
BOM	Bill of material

6.4 Information sources

Books:

EMC and the printed circuit board : design, theory, and layout made simple / Mark I. Montrose (available in college library) This book covers a bit about board layout, it is mainly focused on suppressing noise and radiation from PCB.

Internet:

www.farnell.com	Farnell, electronic component distributor, datasheets
www.mini-circuits.com	RF component manufacturer
www.philipslogic.com	manufacturer of majority of components used
www.sss-mag.com	webzine about spread spectrum
www.tapr.org	another webzine about spread spectrum
www.interq.or.jp/japan/se-inoue/e_eagle.htm	Eagle tutorial

7 Conclusions

7.1 Final comment about system and documentation

I tried to assembly complete documentation that would enable to easily reproduce the system. Unfortunately as it is usual in all project time is our enemy and deadline for handling results is uncomfortably close. Due to this reason, there are few issues left incomplete. After I constructed final version of board I discovered there were few more component needed, so they are usually just added on the board without rebuilding whole board. I tried to correct these mistakes in schematic so that the schematics are complete. However I did not introduce these changes to board artwork so the artwork need a bit revising and rerouting before they can be used. On final boards there were also few short circuits and missing tracks. Missing track are usually cause by fact that in prototyping I used many test points and wire connections and afterward I forgot to remove them from final version. These connections are usually done in layer 5, 6 or 7. I overcome those by using wire connections or scratching out the copper where necessary, but again for reproduction, artwork should be revised for these errors.

7.2 Legal Issues

It is quite a paradox that even when I was working on official college project I managed to break two legal issues. First is usage of illegal software as mentioned above. The other one is illegal unlicensed transmission. According to law it is fine to build transmitter of any kind, but only as long as you do not connect antenna. So by connecting my lovely dipole antennas to my system I committed criminal offence. Consulting frequency allocation chart for Ireland I found that on frequencies around 70MHz where I generated most of the garbage should be some government services and aeronautical navigation, so I only hope that I did not intercept some super important super secret government transmission or did not caused plane crash and nobody is going to prosecute me for that. Moreover transmitting power was relatively small, so area affected should not be larger than few kilometers around the college.

7.3 College support

I have to say that it was pleasure doing the project on CIT. Students on CIT generally enjoys more freedom than I am used from my home institution (Czech Technical University, Prague). Absolutely wonderful thing is unlimited access to college labs facilities with good equipment that I really enjoyed a lot and without what my project probably would not be successful. Basically I usually always get what I asked for. I have virtually unlimited access to appropriate equipment needed for measuring and testing. There were no problems with financing and ordering components so I really did not have any worry about component usage. Only problem I encountered was above mentioned problem with PCB design software. Cooperation with staff

member was good and they always tried to help me as only as they could, unfortunately it was not always enough ☺ as nobody on the college has any experience with system I was building (maybe except Mr. Brian Donovan).

7.4 Impression from the project

Now when project is finished I see what I have done wrong and that nearly everything could be done more efficiently. When I look back and realize that this cost me 7 month of intensive work, usually something like 10 hours in the college nearly every day, I have impression that result is really poor. If somebody asks me how would I characterize my project, I usually say that was funny playing with components and expensive equipment by one desperate amateur. To be honest I really do not think that there is anything on my project that would even slightly resembles work of engineer. I have no calculation, no simulation and no professional looking result. My board is piece of crap, which would not pass any EMC testing and which works only if it is in good mood. On the other hand, after few words with guy from Analog devices, I found that it is virtually impossible for individual like me to create any professional product. Big companies usually uses internally developed simulation programs with own models and have several very experienced engineers who help with the design. Also supporting facility for design engineers are probably a bit different than one I can get on the college where I have to do all work myself. Anyway I really enjoyed this project and I think it gave me a lot of practical experience connected with constructing real circuit and I only hope it did not affect my sanity as now I can tell resistor value only by looking on color code ☺.

8 Appendix A – Original article, concept for my project

Voice Link Over Spread Spectrum Radio

by James A. Vincent, G1PVZ (g1pvz@tapr.org)

Originally published in **ELECTRONICS WORLD + WIRELESS WORLD** (A Reed Business Publication) September/October 1993

Until quite recently spread spectrum techniques were almost exclusively in the military domain. Their use in GPS and the latest cellular phones will be followed by many other civil applications. This article, the first of [two] parts, examines the technology by describing an experimental direct sequence voice transmission system as a worked example.

8.1 1:Basic Principles

Most Communication Engineers are used to minimising transmission bandwidths. The trend has been to use narrower bandwidths, as with the transition from double sideband to single sideband modulation. It is quite obvious that narrower bandwidths permit more communication channels to be packed into a defined frequency band.

However the rationale of using the very wide bandwidths required by Spread Spectrum systems needs explanation. Claude Shannon produced a ground breaking paper on the mathematical theory of communication in 1949. Shannon's resulting theorem can be expressed as:

$$C = W \log_2 \left[1 + \frac{S}{N}\right] \text{bits}^{-1}$$

where C = data rate in bits per second, W = bandwidth (Hz), S = average signal power (W), N = mean white gaussian noise power (W). It can be seen from the equation that the only options available to increase a channel's capacity are to increase either the bandwidth (W) or the signal to noise ratio (S/N).

An increase in the signal to noise ratio requires an increase in transmitter power as the noise within the channel is beyond our control! Thus we can either trade power or bandwidth to achieve a specified channel data rate. Because of the logarithmic relationship, increasing the power output is often unrealistic. However if frequency allocation constraints permit, the bandwidth can be increased. An appreciable increase in data capacity or signal to noise ratio (for a fixed data rate) can then be achieved.

Spread spectrum systems utilise very wide bandwidths and low signal to noise ratios. From Shannon's theorem:

$$C = W \log_2 \left[1 + \frac{S}{N}\right]$$

$$\frac{C}{W} = \log_2 \left[1 + \frac{S}{N}\right]$$

changing bases

$$\text{As} \quad \log_a P = \frac{\log_b P}{\log_b a}$$

$$\therefore \log_b P = \log_b a \log_a P$$

$$\frac{C}{W} = \log_2 e \times \log_e \left[1 + \frac{S}{N}\right]$$

$$\text{Now} \quad \log_a b = \frac{1}{\log_b a}$$

$$\frac{C}{W} = \frac{1}{\log_e 2} \times \log_e \left[1 + \frac{S}{N}\right]$$

$$\frac{C}{W} = 1.44 \log_e \left[1 + \frac{S}{N}\right]$$

By logarithmic expansion

$$\log_e \left[\frac{S}{N}\right] = \frac{S}{N} - \frac{1}{2}\left[\frac{S}{N}\right]^2 + \frac{1}{3}\left[\frac{S}{N}\right]^3 - \frac{1}{4}\left[\frac{S}{N}\right]^4 + \frac{1}{5}\left[\frac{S}{N}\right]^5 \ldots \text{etc}$$

In a spread spectrum system the signal to noise ratio (S/N) is typically small, much less than 0.1

$$\frac{C}{W} = \frac{1.44S}{N} \quad \text{thus} \quad W \approx \frac{NC}{S}$$

From the derived relationship it can be clearly seen that a desired signal to noise ratio for a fixed data rate *C*, can be achieved by increasing the transmission bandwidth.

For example, assume a data rate of 32 Kbits^{-1} and a signal to noise ratio of 0.001 (-30 dB)

$$W \approx \frac{CN}{1.44S}$$

$$\text{thus} \quad W \approx \frac{32 \times 10^3 \times 1000}{1.44} \approx 22 \text{ MHz}$$

So for a data rate of 32 Kbits^{-1}, operation at the very low S/N ratio of -30db is achievable by spreading the signal over a bandwidth of 22 MHz. By using a very much wider bandwidth than that of the original data it is possible to maintain data capacity without increasing the transmitter output power. It is an extreme example of a power-bandwidth trade off.

Two criteria (see Dixon) for a spread spectrum system are:

- that the transmitted bandwidth is much greater than the bandwidth or rate of the information being sent; and
- that some function other than the information being sent determines the resulting radio frequency bandwidth.

The two major techniques used in spread spectrum systems are frequency hopping (fh) and direct sequence (ds). Of the two, frequency hopping is perhaps the easiest to visualise. In a frequency hopping system the frequency or channel in use is changed rapidly. The transmitter hops from channel to channel in a pre-determined but pseudo-random sequence (see Figure 1). The receiver has an identical list of channels to use (the hop set) and an identical pseudo-random sequence generator to that of the transmitter. A synchronising circuit in the receiver ensures that the pseudo-random code generator in the receiver synchronises to the one in the transmitter. When the transmitter and receiver are synchronised the user is unaware that the transmitter and receiver are rapidly changing frequency.

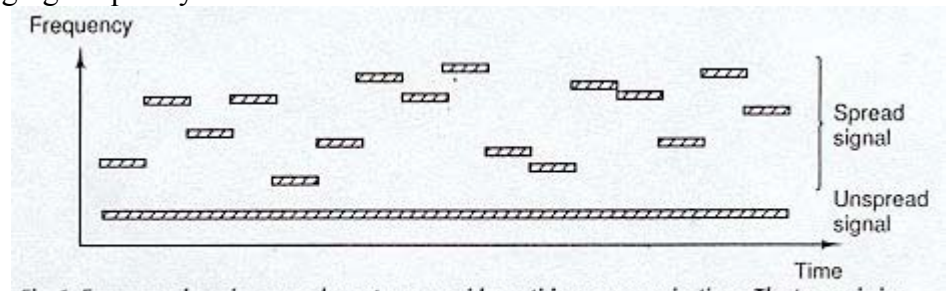


Fig. 1. Frequency hopping spread spectrum provides anti-jam communications. The transmission frequencies change periodically in discrete, pseudorandom steps, the sequence of which is known only by the intended receiver. Multiple users interleave transmissions.

However should the receiver not be synchronised to the transmitter or a conventional receiver be used, nothing will be heard unless the transmitter hops onto the receiver's tuned frequency. As a frequency hopping transmitter typically hops over tens to thousands of frequencies per second (the hop rate), the time it stays on a particular channel (the dwell time) is very short and as a result the signal would appear as a burst of interference.

The other major spread spectrum technique is known as direct sequence or pseudo-noise. In this technique a pseudo-random code directly phase shift keys the carrier increasing its bandwidth (see Figure 2). In a typical direct sequence system a double-balanced mixer (DBM) is driven by the PN code to switch a carrier's phase between 0 degrees and 180 degrees. This is known as biphase shift keying (BPSK) or sometimes phase reversal keying (PRK). Unlike a frequency hopping transmitter where the pseudo-random sequence commands a synthesiser to change frequency, the direct sequence signal is directly generated by the pseudo-random sequence.

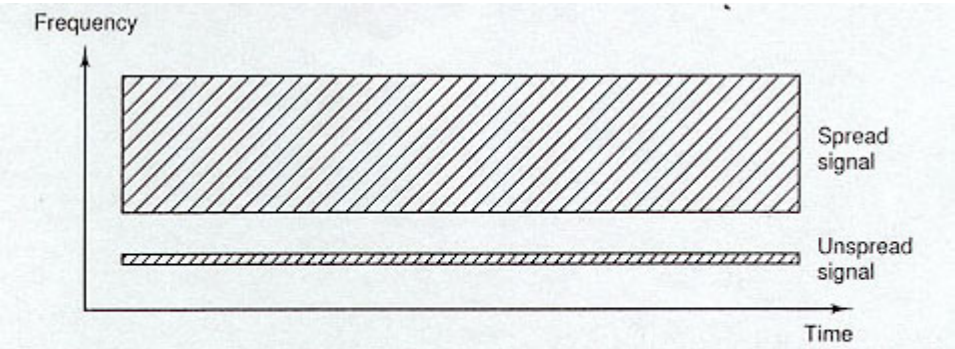


Fig. 2. Direct sequence spread spectrum. An RF carrier modulated with this pseudorandom code will be spread into a wide bandwidth continuous spectrum signal which looks like noise. At the receiver, the same pseudorandom code is used to correlate and hence despread the signal. Many transmissions may use the same frequency simultaneously; signals using different pseudorandom codes do not correlate and hence look like an addition to the background noise.

The receiver despreads this wideband signal by using an identical synchronised pseudo-random code to that in the transmitter. As with the frequency hopper, the receiver must use a circuit to adjust its clock rate so that the receiver's pseudo-random code is at the same point in the code as the transmitter. A tracking circuit is necessary to maintain synchronism once it has been attained.

8.1.1 Sending data with spread spectrum

Spread spectrum signals (whether direct sequence, frequency hopping or their hybrids) can support any conventional analogue or digital modulation scheme to impress data onto the spread spectrum carrier.

Obviously some modulation formats are less suitable than others. Amplitude modulation and its derivatives are the least desirable as their use will destroy the signal's uniform power spectral density. This constant carrier envelope is very desirable for spread spectrum systems designed for covert usage.

Frequency modulation (frequency shift keying for data) is often used in frequency hopping systems, but is infrequently used in direct sequence systems. This is because when a direct sequence signal passes through a squaring or frequency doubling circuit, a carrier at twice the signal's centre frequency is produced. This twice frequency narrowband carrier will contain any modulation impressed on the direct sequence signal. Thus with analogue modulation it is possible for the signal to be demodulated without any prior knowledge of the pseudo-random spreading code.

One of the commonest modulation techniques used in conjunction with direct sequence is known as code inversion or modification. The digitised voice or digital data is exclusive ORed with the PN spreading code. This will invert the PN code sequence if the data is a "1" or pass the PN code unmodified if it is a "0". Provided that the data stream is synchronised with the PN code, the correlation properties of the code are unaffected.

Assuming synchronisation at the receiver, the unmodified code despreads the direct sequence signal. This produces a narrowband signal which is still biphase shift modulated, but this time with the data or digitised speech. This signal can then be demodulated by a conventional biphase shift demodulator such as a squaring or Costas loop demodulator.

This code modification modulation is simple to implement in the transmitter and relatively easy to demodulate in the receiver. It also has the advantage of providing message privacy which the analogue modulated direct sequence signal does not have. It should be noted that it is possible to directly demodulate uncorrelated spectral components of an analogue modulated direct sequence signal should the demodulating receiver be very close to the transmitter. In addition the code modification technique preserves the constant power envelope of the direct sequence signal.

One disadvantage of code modification is that voice or other analogue signals require digitisation. As in any system design, the selection of the digitisation technique is very important. The technique selected must use the lowest possible data rate as data rate is inversely proportional to the process gain of the system. The technique selected for the system described uses a enhanced form of delta modulation to digitally encode the voice into a serial data stream.

8.1.2 Delta modulation

Delta modulation is a variation of pulse-code modulation. It compares successive signal samples and transmits only their differences, rather than the actual amplitude as in PCM. This reduces the number of bits required to code the speech. The continuous audio signal is sampled at periodic intervals. The sampled value is then compared with a staircase approximation of the output signal. If the sampled waveform exceeds the staircase approximation, a positive pulse is generated. If the sampled waveform is less than the staircase approximation, a negative pulse is generated. This output pulse, positive or negative, forms the next step in the staircase approximation, i.e. the sum

of the binary pulse train at the output of the encoder produces the delta-modulated waveform (see Figure 3).

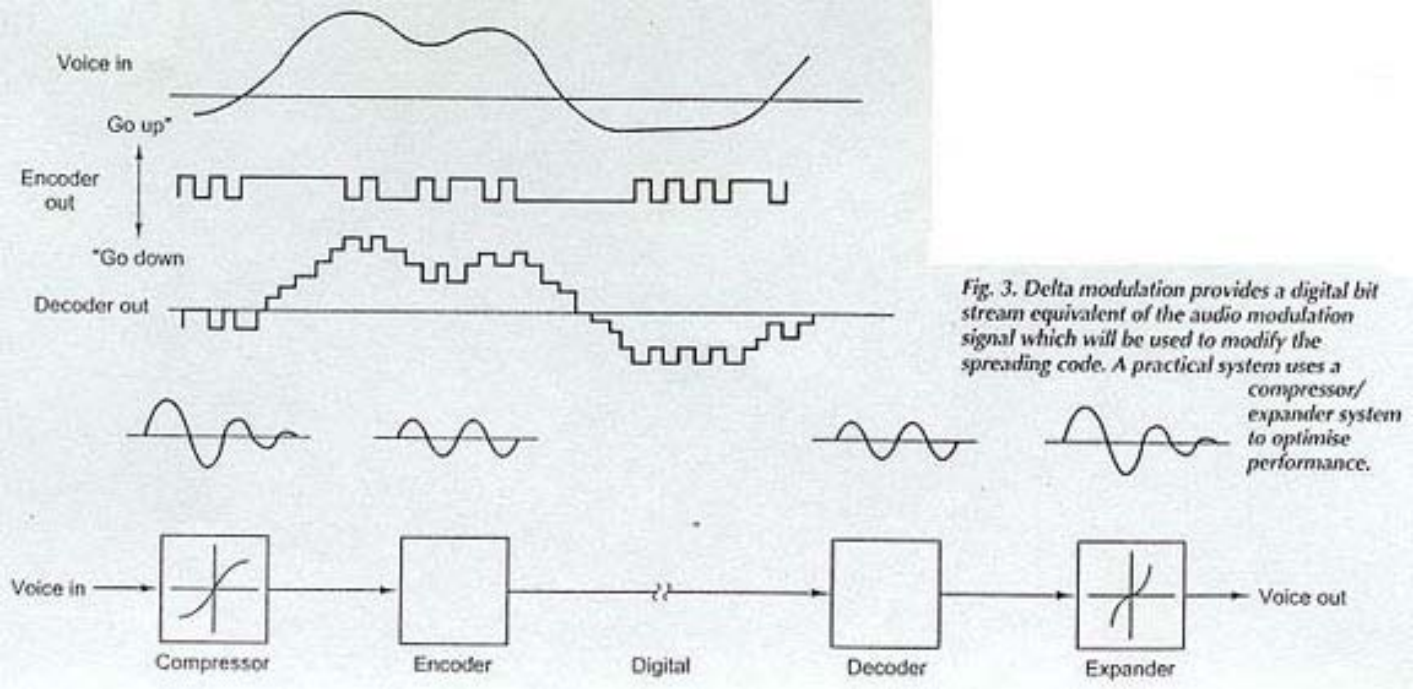


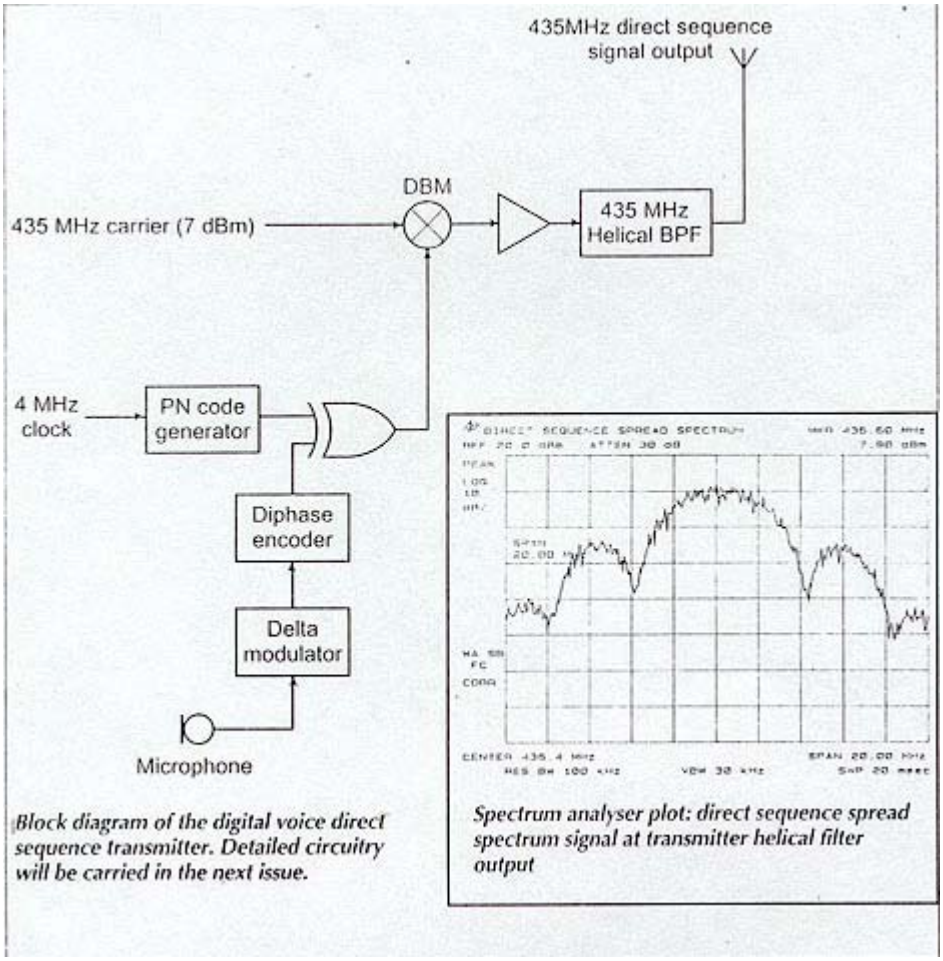
Fig. 3. Delta modulation provides a digital bit stream equivalent of the audio modulation signal which will be used to modify the spreading code. A practical system uses a compressor/ expander system to optimise performance.

At the receiver, the transmitted pulses are integrated and passed through a low-pass filter to remove unwanted high frequency components. The output consists of the original analogue signal together with some additional noise somewhat similar to quantisation noise.

Continuously Variable Slope Delta Modulation (CVSD) takes advantage of the fact that voice signals do not change abruptly and that there is only a small change from one sample to the next. A reasonably good reproduction can be obtained by transmitting in a given interval whether the output signal should increase or decrease. A linear delta modulated system has the undesirable feature that there is one input level which maximises the signal to noise ratio. In CVSD this is overcome by compressing the large amplitude in the signals relative to the smaller ones prior to encoding using a compressor circuit. In this way the input level to the encoder can be maintained close to the value which gives the maximum signal to noise ratio.

The receiver decodes the delta modulated binary stream and passes the analogue signal through an expander to counteract the effects of the transmitter compressor. Companding is optimised for the human voice. CVSD is considerably more effective than standard delta modulation and also exhibits less serious sound degradation in the presence of digital noise interference than PCM.

8.1.3 Circuit description



Block diagram of the digital voice direct sequence transmitter. Detailed circuitry will be carried in the next issue.

Spectrum analyser plot: direct sequence spread spectrum signal at transmitter helical filter output

The system is described in functional blocks. First, the transmitter direct sequence modulator. The exciter's clock frequencies are provided by a master 4 MHz crystal oscillator and a divider. Power-up reset (with manual override) is configured around a Schmidt-trigger.

A shift register and exclusive OR gates are configured as a 4 MHz 127 chip (code bit) long maximal pseudo-random code generator (see section *pseudo-random codes and generation*).

Microphone audio is amplified by the VOGAD (Voice Operated Gain Adjusting Device) to the optimum level for the input of the delta modulator. The delta modulator converts the audio into a 32 $Kbits^{-1}$ serial data stream (see column sending data with spread spectrum). This serial binary data stream must be coded into a format which is polarity insensitive because the receiver demodulator cannot recover the de-spread data's absolute phase. Only data transitions are recovered at the receiver, hence there is no way of determining whether the output data stream is inverted or not.

The digitised audio is converted from a non return to zero (NRZ) format into a polarity insensitive diphase (biphase-mark) data stream. This sub-circuit produces a diphase signal (Figure 4), where a logic 1 has start, mid-bit and end transitions and a logic 0 has only start and finish transitions.

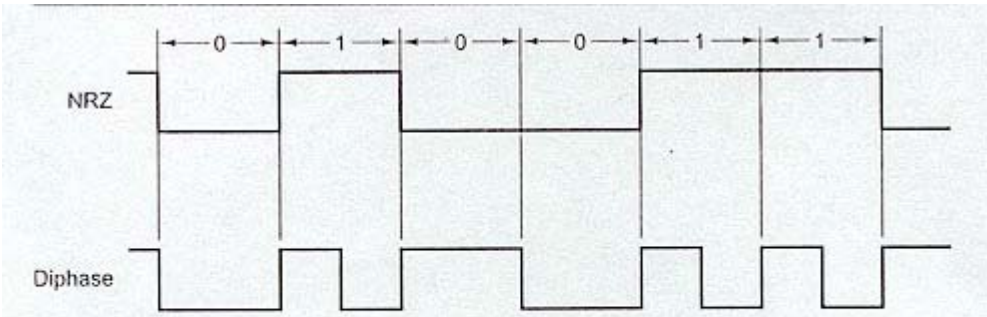


Fig. 4. The transmitted signal must be insensitive to phase. Digitised audio in phase-sensitive NRZ is converted to its diphase equivalent for transmission enabling the receiver to recover the data with the correct polarity. This diphase code is exclusive-ORed with the pseudorandom spreading code to provide a drive signal for the double balanced modulator of the sort shown below.

In addition to providing phase insensitive data transmission the format also makes clock recovery at the receiver relatively easy, as unlike NRZ even a continuous stream of diphase encoded 0's results in many start and finish data cell transitions. The diphase encoded delta modulated digital voice signal is ex-ORed with the pseudo-random code producing a code modified PN spreading code.

The data modified PN code from the output of the exclusive-OR gate provides a balanced drive (plus or minus 24 mA as an *AC* logic family device has equal sink and source currents) via a coupling capacitor and 50 ohm matching pad, to a double balanced mixer (DBM) configured as a biphase shift keyer.

The PN code output alternately sinks and sources current, causing the diodes in the DBM to alternately switch on and off producing 180 degree phase reversals in the 435 MHz carrier signal (see Figure 5). The output spectrum consists of a series of symmetrical sidebands which have a $Sinc^2x$ distribution due to the many frequency components of the pseudo-random code.

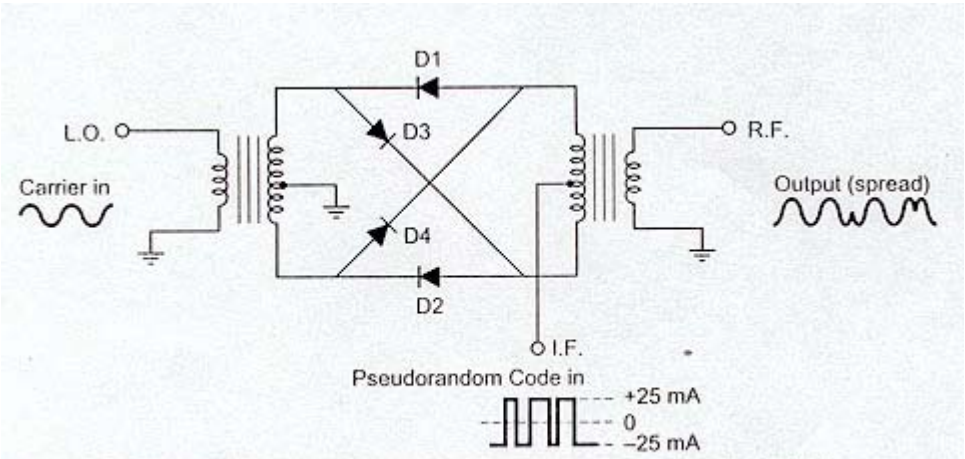
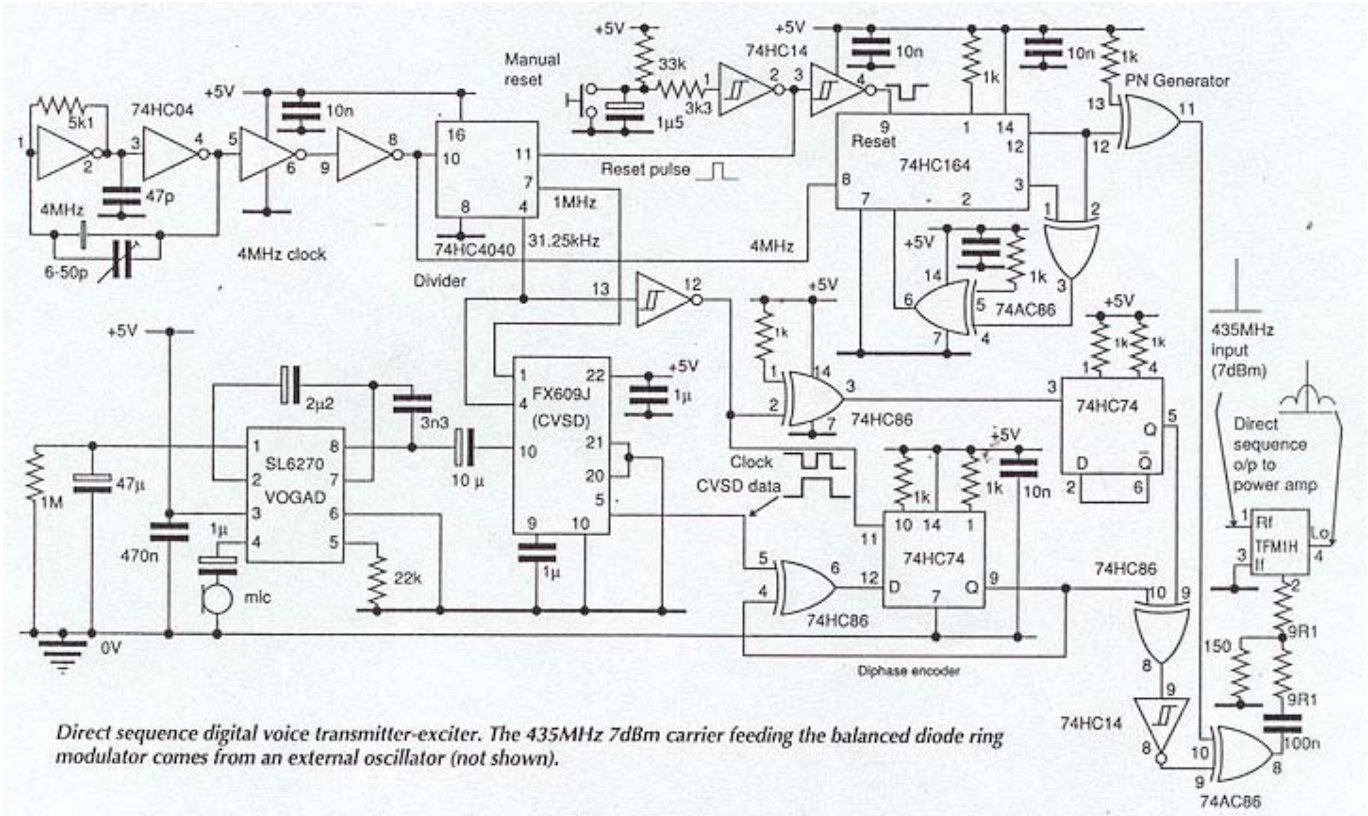
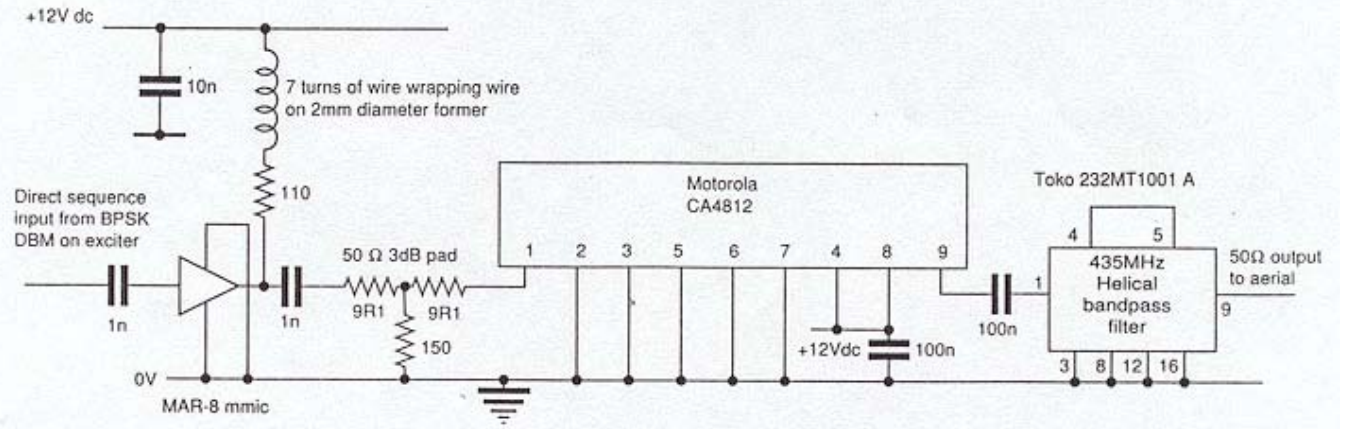


Fig. 5. The biphase modulator. Diode pairs D_1, D_2 and D_3, D_4 alternately switch with the pseudorandom code signal causing phase reversal between the local oscillator and output ports

As the spreading code has a pseudo-random character, the occurrence of a particular frequency is pseudo-random in time and the direct sequence output appears as noise on a spectrum analyser. The spread spectrum signal has a main lobe bandwidth of 8 MHz (twice the PN code clock rate for BPSK). This is amplified by a *MAR8* MMIC (monolithic microwave integrated circuit) and further amplified to around 100 mW by a Motorola *CA4812* Class A Amplifier module. Helical band pass filtering is used to ensure that the output signal is within the permitted bandwidth before free-space transmission.



Direct sequence digital voice transmitter-exciter. The 435MHz 7dBm carrier feeding the balanced diode ring modulator comes from an external oscillator (not shown).



Power amplifier and driver circuit for direct sequence transmitter.

8.1.4 Spread spectrum terminology

Process Gain (Gp) is a fundamental concept in spread spectrum systems. The process gain indicates the gain or signal to noise improvement exhibited by a spread spectrum system by nature of the spreading and despreading process. Process gain can be estimated by the following empirical relationship.

$$\text{Process Gain} = \frac{G_p}{R_{info}} = BW_{RF}$$

$$\text{Process Gain} = 10\log_{10}\left[\frac{BW_{RF}}{R_{info}}\right]\text{dB}$$

where,

BW_{RF} = 3dB bandwidth of the transmitted spread spectrum signal (Hz). R_{info} = data rate of the information transmitted (bits per second).

For a direct sequence signal, BW_{RF} is assumed to be equal to the 3 dB bandwidth of the spectrum (which is 0.88 times the pseudo-random code clock rate for a biphase shift keyed direct sequence system). For a frequency hopping system BW_{RF} is equal to *m* times the channel bandwidth where *m* is the number of frequency channels available.

Jamming Margin. Although the process gain is directly related to the interference rejection properties a more indicative measure of how a spread spectrum system will perform in the face of interference is the jamming margin (*Mj*). The process gain of a system will always be greater than its jamming margin.

$$M_j = G_p - \left[L_{system} + (S/N)_{out}\right] \text{dB}$$

where,

L_{system} = system implementation losses (dB); G_p = process gain (dB); $(S/N)_{out}$ = signal to noise ratio at the information output. (dB)

A spread spectrum system with a 30 dB process gain, a minimum required output signal to noise of 10 dB and system implementation loss of 3 dB would have a jamming margin of 30-(10+3) dB which is 17 dB. The spread spectrum system in this example could not be expected to work in an environment with interference more than 17 dB above the desired signal.

Power Spectral Density. By nature of the spreading process, the output power of the spread spectrum transmitter is spread over typically many megahertz of bandwidth. The spectral density is the number of Watts of radio frequency power present per Hertz of bandwidth. Thus for a direct sequence transmitter of 1W output and a spread bandwidth of 8MHz the power spectral density is:

$$\frac{1}{8{,}000{,}000}\text{W / Hz} = 125\,\text{nW / Hz}$$

For a conventional AM transmitter, power spectral density is around

$$\frac{1}{6000}\text{W / Hz} = 166\ \mu\text{W}$$

some 31dB greater.

The advantage to the military user is that the signal strength apparent to a conventional narrowband receiver is very, very low and would probably not be recognised as a communications signal, hence the expression "Low Probability of Intercept" and "Low Probability of Recognition".

8.1.5 Glossary

Antijamming (AJ): Techniques used to minimise the effects of jamming or unintentional interference.

***Auto-correlation*:** This is a measure of similarity between a signal and a time shifted replica of itself. Auto-correlation is a special case of cross-correlation. The auto-correlation function is the fundamental theoretical basis of spread spectrum communications.

Biphase Shift Keying (BPSK): A phase shift keying technique where the carrier phase changes between 0 degrees and 180 degrees (0 and pi radians) under the control of a binary code. BPSK is frequently used to generate direct sequence spread spectrum signals, where the binary code is a pseudo-random sequence.

Chip: A single element of the spreading code. This may be one or more of the PN code bits, depending on the modulation technique used. For BPSK one chip represents one code bit, whereas for quadrature phase shift keying (QPSK) one chip represents two code bits.

This is because there are four states for QPSK (0, 90, 180 and 270 degrees) and only two states for BPSK (0 and 90 degrees). Obviously two binary bits are required to represent four states and only one bit for two states.

Code: The term code usually refers to the pseudo-random code used to control the modulation technique used to spread the carrier.

Code Division Multiple Access (CDMA): A multiplexing technique where each user is given a different pseudo-random spreading code. To communicate with a particular user, the sender must select the code assigned to that user.

If the CDMA codes are carefully selected to ensure good correlation properties, then unwanted CDMA transmissions will not be correlated and hence rejected as wideband interference (up to the limit of the jamming margin M_j of the system). This technique can permit many users to operate simultaneously on the same frequency.

Correlator: A device to measure the similarity of two signals. Sometimes referred to as a de-spreader in direct sequence systems.

Costas Loop: A compound phase locked loop sometimes called an I-Q (In-phase/Quadrature phase) loop. It is used for demodulating double-sideband suppressed carriers (DSBSC) which is the modulation format of a biphase phase-shift keyed signal.

Cross-correlation: This is a measure of the similarity of two signals.

Delay Locked Loop: A tracking circuit which ensures the direct sequence receiver PN clock tracks (follows) any variation in the transmitter's PN clock rate once synchronisation has been achieved. (See column The Delay Locked Loop).

Delta Modulation: A analogue to digital conversion technique (see column Sending Data with Spread Spectrum).

Diphase (biphase-mark): A polarity-insensitive waveform, where a transition occurs at the beginning of every data period. A logic 1 is represented by a transition one half period later. There is no second transition for a logic 0.

Direct Sequence (ds): A spread spectrum modulation technique where a pseudo-random code directly phase modulates a carrier, increasing the bandwidth of the transmission. The resulting

signal has a noise-like spectrum. The signal is despread by correlating with a pseudo-random code identical to and in synchronism with the code used to spread the carrier at the transmitter.

Frequency Hopping (fh): A spread spectrum modulation technique where the transmitter frequency hops from channel to channel in a predetermined but pseudo-random manner. The signal is de-hopped at the receiver by a frequency synthesiser controlled by a pseudo-random sequence generator synchronised to the transmitter's pseudo-random generator.

Jamming Margin (M_j): A measure of a spread spectrum system's resistance to jamming or un-intentional interference, (see column Spread Spectrum Terminology).

Linear Codes: Pseudo-random codes generated using only modulo-2 addition or subtraction,(see column Pseudo-random Codes and their Generation).

Maximal Code: A maximal code is the longest that can be generated with a feedback type pseudo-random generator (see column Pseudo-random Codes and Generation).

Process Gain (G_p): The measure of the gain or signal-to-noise improvement exhibited by a spread spectrum system by nature of the spreading and de-spreading process.

Pseudo-noise: Code sequences which have noise-like properties. The term pseudonoise (pn) is often used for direct sequence systems which use such codes to spread the carrier.

Sinc x: Sinc x is the mathematical term for the following expression:

$$\operatorname{sinc} x = \frac{\sin x}{x}$$

A BPSK spread spectrum has a Sinc^2x power spectrum.

Squaring Loop: A BPSK (or DSBSC) demodulator which regenerates the suppressed carrier through a frequency squaring (or doubling) process. This doubling process produces a twice frequency unmodulated carrier, which when divided by two can be multiplied with the input BPSK signal to recover the data.

8.2 2: Detailed Circuitry

Pseudo-random codes can be categorised as being linear or non-linear codes. Linear codes are generated using linear operations (which for binary pseudo-random codes is solely modulo-2 addition or subtraction). This essentially means only ex-OR gates are used in the shift register feedback path. A pseudo-random generator which does not use such techniques is termed non-linear.

The most commonly used group of pseudo-random sequences used in spread spectrum are the maximal linear code sequences (sometimes called M-sequences or PN -- pseudo-noise -- codes). Maximal codes are the longest codes that a shift register of specified length can produces and have mathematical properties well suited to spread spectrum communications.

A maximal shift register pseudo-random generator consists of a shift register with selected outputs being exclusive-ORed and fed back into the shift register input. The circuit goes through a number

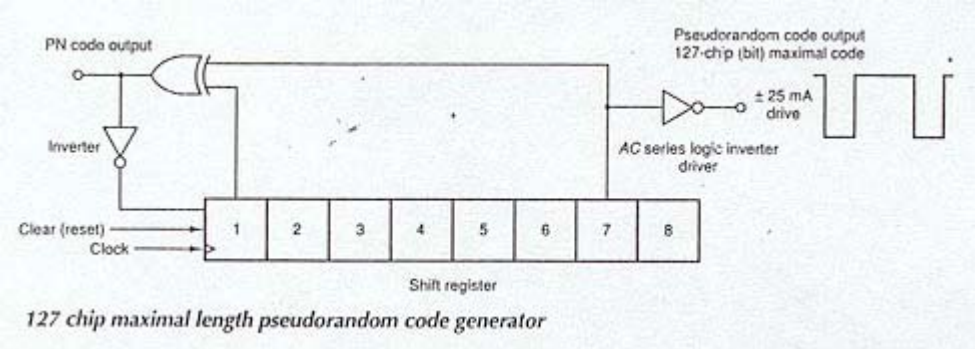
of states (determined by the bits in the shift register at each clock pulse) before it repeats itself after a set number of clock pulses. The maximum number of states for a shift register of length *m* is 2^m, ie for a 7-stage shift register $2^7 = 128$ states. However the all-zero state is not allowable as the pseudo-random generator would lock-up as ex-ORing two logic 0 results in yet another logic 0 at the input. Therefore a maximal length pseudo-random code generator can produce a pseudo-random sequence 2^m-1 bits long before repeating itself.

To obtain a maximal sequence, the correct shift register outputs (tap points) must be found. These could be found be experimentation but this would be very time consuming! However tables of feedback connections are available.

A 7-stage (ie seven flip-flop) shift register can produce a maximal code of length 2^7-1 = 127 bits (known as *chips* in spread spectrum terminology) long. The feedback tap points may be taken from the following stages:

```
[7,1] [7,3] [7,3,2,1] [7,4,3,2] [7,6,4,2]
[7,6,3,1] [7,6,5,2] [7,6,5,4,2,1] and
[7,5,4,3,2,1]
```

As the simplest circuit implementation is often desired, the first option of tapping the seventh and first stages is selected.



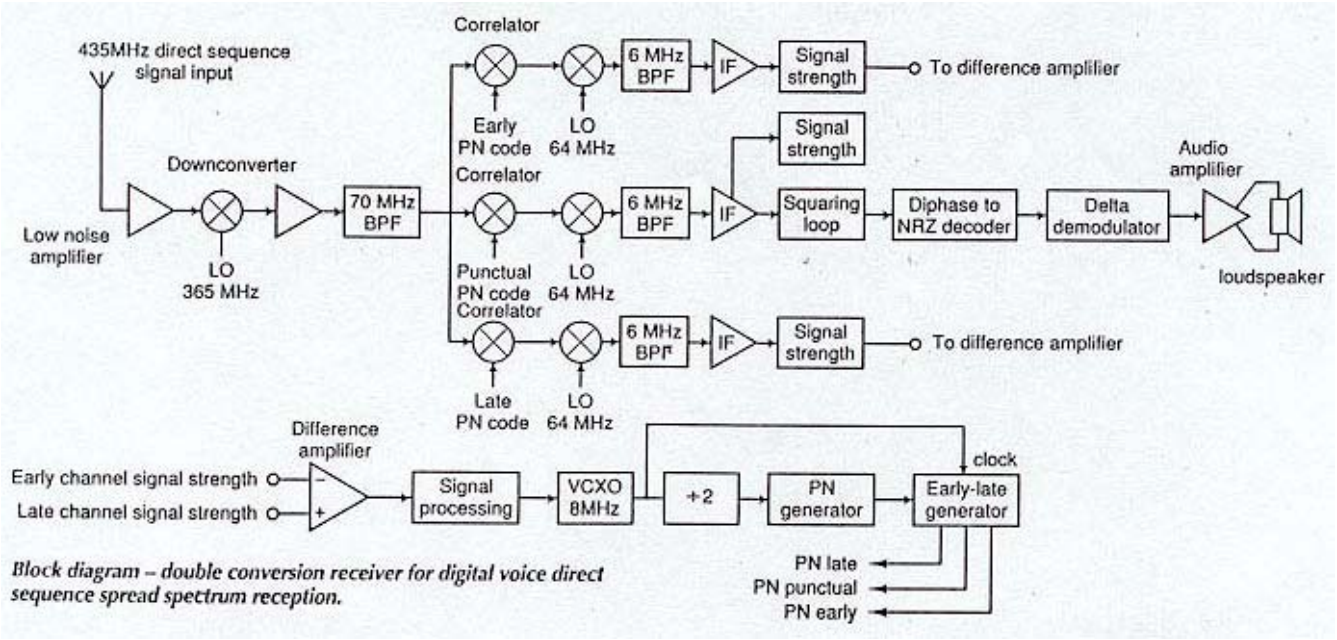
127 chip maximal length pseudorandom code generator

To avoid the all zero-lock up problem, inverting stages are inserted before the shift register input and at the output of the shift register. When the shift register is switched on, a reset pulse is initiated. This pulse initiates all shift register outputs to logic 0. This would normally lock up the pseudo-random sequence generator. However the input inverter injects a logic 1 so that the maximal sequence can commence. The output inverter ensures that maximal code output is inverted negating the effect of the anti-lock-up inverter at the input. The maximal code is also available at the output (A) of the modulo-2 adder, but the second inverter output is normally used to permit direct drive of the DBM in a direct sequence system.

Receiver Functional Description

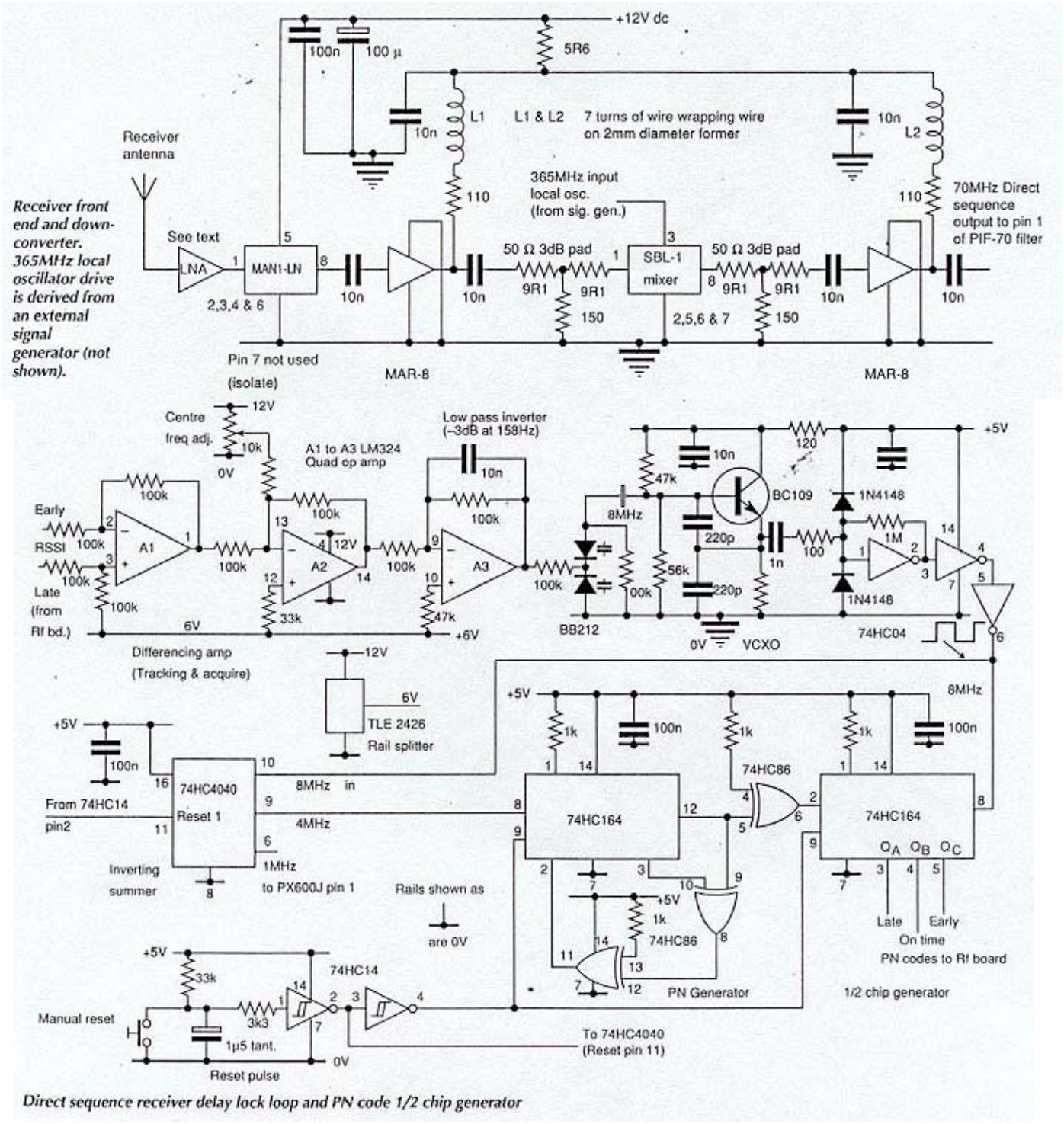
The 435 MHz direct sequence (ds) signal is first amplified by a low noise amplifier[2] followed by a helical filter and further amplification by a low noise amplifier block (*MAN1-LN*) and a *MAR8*. The ds signal is mixed with a 7 dBm 365 MHz local oscillator in a down-converter. The ds signal now centred on an intermediate frequency of 70 MHz is amplified (*MAR8*) and bandpass filtered (*PIF-70*), before being resistively split into three identical signal paths. In each signal path (late, on time or early) the 70 MHz signal is amplified by a further *MAR6* amplifier. A DBM configured as a biphase shift keyer is driven by an early, on time or late pn code. The DBM is buffered by 50 ohm pads and driven by an AC logic buffer as with the DBM used as a BPSK modulator in the transmitter.

Assuming synchronism the despread output is injected into a *NE605* low power FM IF integrated circuit. The second local oscillator at 64 MHz in conjunction with the on-chip mixer downconverts the despread signal (which contains the data in a BPSK format) to 6 MHz. The *NE605* further amplifies the 6 MHz signal and provides filtering using 6 MHz ceramic filters originally designed for television sound strips.



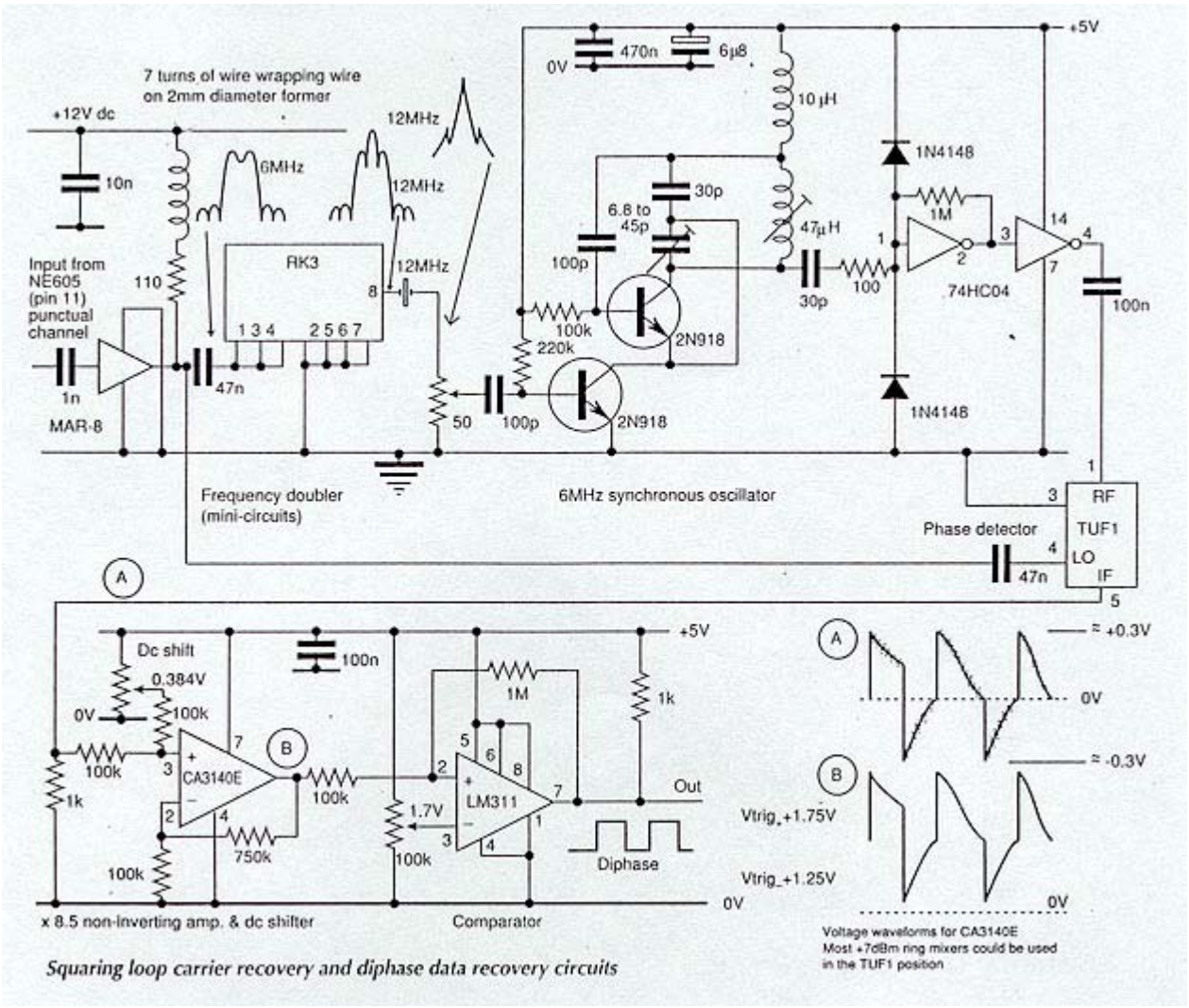
Block diagram – double conversion receiver for digital voice direct sequence spread spectrum reception.

A RSSI (received signal strength indicator) is available from each NE605 with a 90 dB range logarithmic output. The RSSI outputs from the early and late channels go to the delay locked loop circuit. The despread data output from the on time (punctual) channel is further amplified by a MAR8 amplifier before being frequency doubled in a Mini Circuits RK3 doubler. As previously discussed the despread data signal has a biphase shift keyed (BPSK) format. The BPSK frequency spectra is similar to that of a double sideband suppressed carrier and as for DSBSC, carrier recovery is required to demodulate the signal. It can be shown mathematically[3] that by squaring or doubling a BPSK signal a twice frequency carrier is obtained. After passing the doubled signal through a 12 MHz crystal used as an exceptionally narrow bandpass filter, the signal is applied to a synchronous oscillator. This versatile circuit (see section *The Synchronous Oscillator*) free runs at 6 MHz and on application of the 12 MHz signal synchronously locks to half of the input frequency, effectively regenerating the 6 MHz carrier reference. This locked 6 MHz output is buffered and amplified to produce a logic level 0, +5v output, which together with the signal output from the on-time (punctual) *NE605* IC is injected into a DBM configured as a phase detector. The voltage output of the phase detector is amplified, level shifted and using a voltage comparator converted into 0, +5v logic levels.



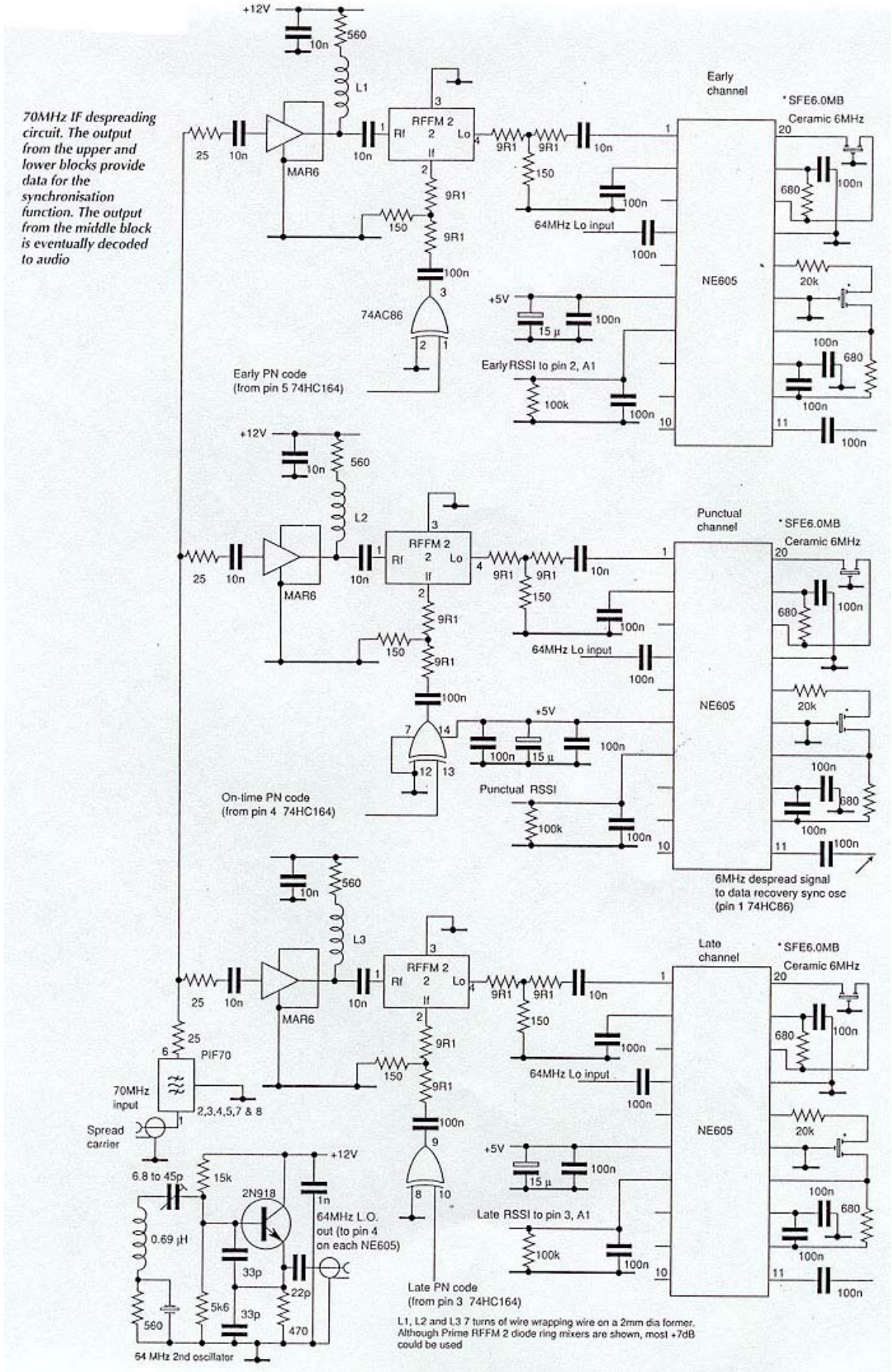
Receiver front end and down-converter. 365MHz local oscillator drive is derived from an external signal generator (not shown).

Direct sequence receiver delay lock loop and PN code 1/2 chip generator



Squaring loop carrier recovery and diphase data recovery circuits

70MHz IF despreading circuit. The output from the upper and lower blocks provide data for the synchronisation function. The output from the middle block is eventually decoded to audio

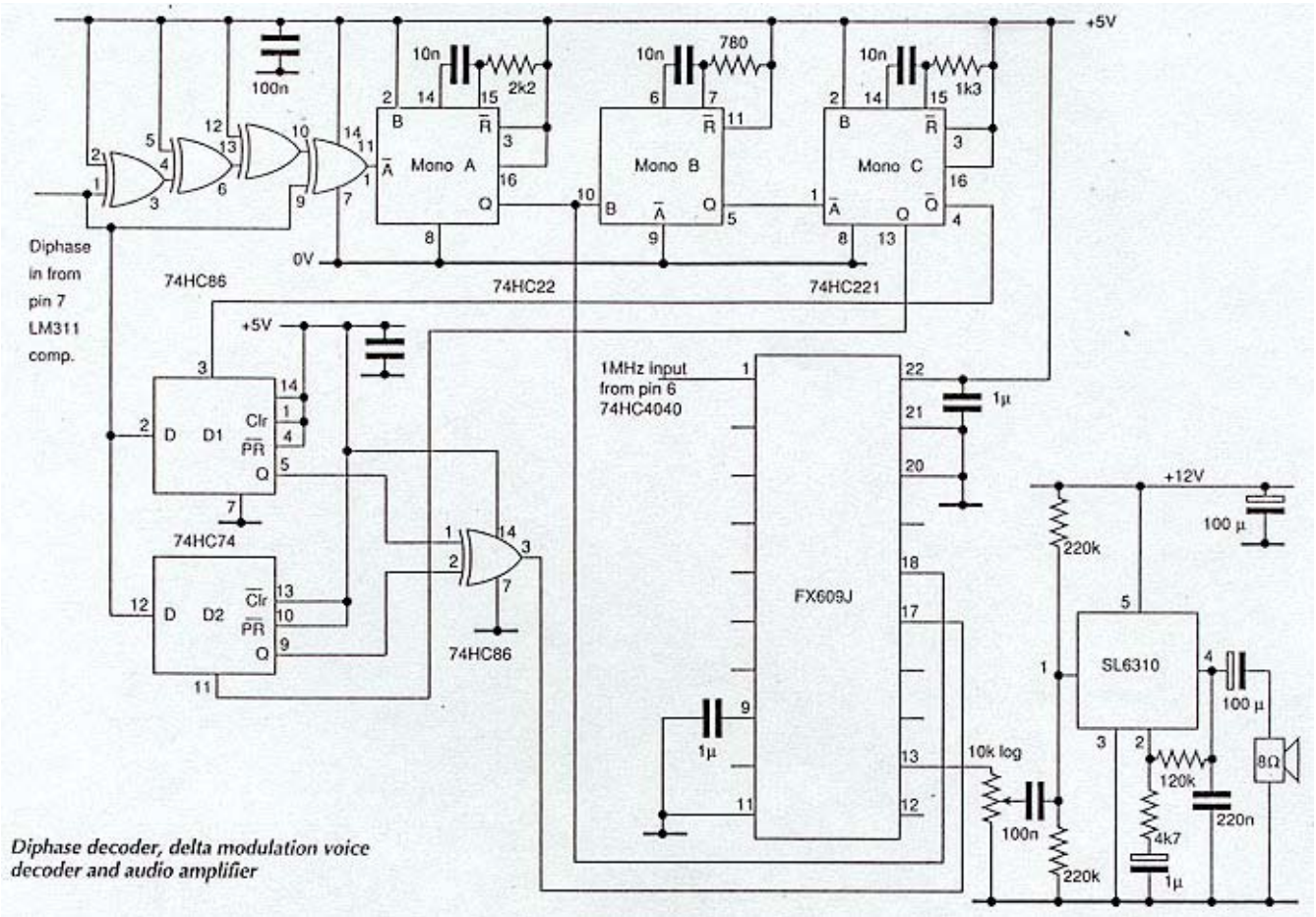


The output from this squaring loop BPSK demodulator does not recover the original data polarity as the original phase of the signal is lost in the doubling process. This is why the data was diphase encoded at the transmitter so that the correct data polarity could be recovered at the receiver.

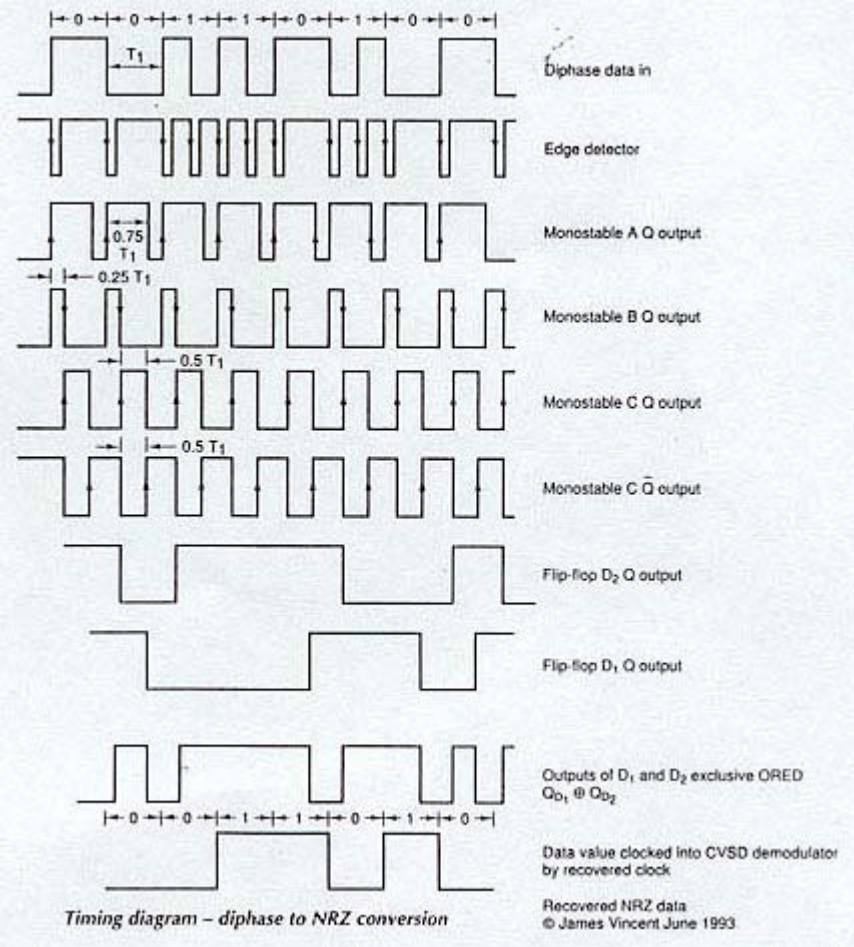
An edge detector configured from an exclusive-OR gates produces a negative pulse for both positive and negative edges of the comparator's diphase data stream output. The edge detector output triggers monostable *A*, one half of a dual monostable. (Note: all monostables are non-retriggable). Monostable *A* is set to produce a positive output pulse with a duration of 75% of the diphase bit cell period. The Q output of monostable A triggers monostable *B* which produces a positive output pulse of duration 25% of the diphase bit cell period.

In turn the negative going edge of monostable *B* output triggers monostable *C* which produces a positive pulse with a duration of 50% of the diphase bit cell period. D-type flip-flop D_1 is clocked by the /Q output of monostable *C* and flip-flop D_2 by the Q output.

The positive edge of the Q and /Q outputs of Monostable *C* occur before and after any mid-bit transition. Thus when D_1 and D_2 are clocked, their outputs will be different if the diphase encoded bit represents a one, or the same if the diphase encoded bit represent a zero. If D_1 and D_2 outputs are exclusive OR-ed then the instantaneous NRZ data is obtained. The clock is recovered at the Q output of Monostable *A*. It can be seen that missing or corrupted diphase data could cause monostable *A* to trigger on a mid-bit transition rather than a 'start' transition. This false synchronisation will be corrected on the next diphase encoded zero as monostable *A* will not be triggered.



Diphase decoder, delta modulation voice decoder and audio amplifier



Timing diagram – diphase to NRZ conversion

The recovered clock and NRZ data is delivered to the delta modulator integrated circuit where it is converted back into audio and amplified to a loudspeaker.

The delay lock loop and code generation circuitry permits code correlation, synchronisation and tracking. The difference amplifier has its inverting and non-inverting inputs respectively connected to the early and late channel RSSI outputs. The difference amplifier is followed by a summing amplifier used to adjust the quiescent frequency of the voltage controlled crystal oscillator and a low pass filter. The output of the inverter drives the control input of the voltage controlled oscillator. The VCXO consists of a high stability AT cut crystal in a discrete transistor based oscillator with varicap frequency control. The oscillator's low voltage output is amplified by approximately 10,000 by the linear biased *HC* logic gate. This hard limits the buffer's output to standard logic levels. The VCXO provides a highly stable, repeatable output which has a 2 kHz tuning range centred on 8 MHz for a tuning voltage of 0 to 6V.

The VCXO output is divided by two to produce a 4 MHz clock. This clock signal drives the 127 chip maximal PN generator. The output of this PN generator is re-clocked through a shift register by the original 8 MHz clock. By extracting the three outputs from neighbouring outputs three identical PN codes are available (early, on-time and late) but with a half clock cycle difference between them. Thus the early code is one clock cycle (or "chip" in spread spectrum terminology) ahead of the late code. Each PN code generator output drives the relevant correlator (de-spreader). See section *Delay Locked Loop*.

In operation the VCXO is offset to a slightly higher frequency than the crystal clock in the transmitter, effectively producing a sliding correlator. Assuming that the receiver is in range and unsynchronised, the receiver code will slide past the transmitter code. At one point in time the two

codes will match. This will result in correlation and the direct sequence signal will be despread. The early channel will be despread before the late channel and the early RSSI value will be considerably higher than the late uncorrelated channel. This difference signal after filtering steers the VCXO output towards the frequency of the transmitter clock. When the receiver and transmitter clocks and PN codes are synchronised the RSSI outputs from the early and late channels will be identical and the difference amplifier output will be zero. Should the receiver clock be retarded greater energy will in the late channel than the early channel, and the VCXO will be driven by the difference amplifier to increase its frequency. If the receiver clock is advanced greater energy will be in the early channel than the late channel and the VCXO will be driven by the difference amplifier to decrease its frequency. Thus the delay locked loop will maintain synchronism once the sliding correlator has caused the receiver to lock. The frequency offset is selected such that it will cause rapid synchronisation but remain within the capture range of the loop.

Construction and Testing

The direct sequence transmitter and receiver were constructed on a combination of Veroboard and double-sided printed circuit boards. The radio-frequency circuits were built on the double-sided pcbs, with the usual RF design techniques employed. The photograph of the completed transmitter-exciter and receiver shows the combination of construction techniques used. (See August *EW+WW*).

The system is designed around easily obtainable components and all inductors and filters are selected from either the Toko or Mini-Circuits range to avoid the difficulties of winding coils.

The set-up of the receiver requires a functioning exciter as a source of a direct sequence signal, hence the exciter is adjusted first. This involves setting the master 4 MHz crystal oscillator with the aid of a frequency counter.

Now the receiver can be directly connected (with a suitable attenuator in-line to prevent overload) to the exciter output. Initially the 64 MHz second local oscillator should be adjusted on frequency. The VCXO's frequency is set using the centre frequency adjust potentiometer, to be slightly higher or lower than twice the measured frequency of the transmitter's master clock.

The resonant circuit of the synchronous oscillator (SO) has to be adjusted until it free-runs at 6 MHz. It is important to ensure that the SO oscillates at 6 MHz (ie not a harmonic) and the input level potentiometer is set to the minimum input level which permits reliable operation.

The gain and comparator reference point potentiometers should be adjusted such that the phase detector recovers the diphase data stream with HC logic compatible levels.

The VCXO frequency is slowly adjusted, until the sliding correlator and delay locked loop lock to and track the transmitter. If a spectrum analyser is available, a narrowband despread BPSK data signal will can be detected at the input to the *PIF-70* filter. A dual channel oscilloscope can be used to monitor and compare the transmitter and receiver (punctual) PN codes. If the receiver has synchronised then the two PN codes will line up and the receiver code will be seen to track the transmitter's code. If all is correctly adjusted then the synchronous oscillator will regenerate the 6 MHz carrier with the data recovery circuit and delta-modulator i.c. recovering the audio. Various waveforms are shown on the circuit diagram to aid trouble shooting.

After this initial procedure the power and pre-amplifiers can be added for free-space checks (provided radio regulations permit). Some minor adjustments particularly of the VCXO frequency

may be required to ensure reliable acquisition and locking. If the VCXO frequency offset is too great then the receiver will initially acquire the signal , but will be unable to track it. A degree of trial and error may be necessary to arrive at a receiver clock offset which provides rapid synchronisation and reliable tracking performance. The prototype took less than two seconds from power-up to synchronise and would remain in lock provided the signal was not lost.

The Radiocommmunications Agency (the UK radio regulatory authority) granted special authority to the author to experiment with spread spectrum techniques on the 70 cm band under his amateur radio service licence.

At present the UK amateur radio licence does not permit the use of spread spectrum modulation. It is hoped that in the future the standard UK licence will permit spread spectrum modes of operation as is allowed in the USA by the Federal Communications Commission.

8.3 Auto-Correlation and Cross-Correlation

The main basis of spread spectrum communications is the correlation function, a measure of the similarity between functions. For the autocorrelation function:

$$\Psi_A(\tau) = \int_{-\infty}^{+\infty} f(t) \times f(t-\tau) dt$$

A time dependent function (such as Sin*w*t) is compared with an identical replica time shifted by a magnitude and summed (integrated) for all values of t. This function has a maximum at $\tau = 0$ which shows that (obviously) a function is most similar to itself when it has not been time-shifted. For periodic functions, further maxima appear for a multiple of this period.

The response of the correlation function at other values than $\tau = 0$ determines how well the original function *f(t)* can be found again by variation of the time shift tau. It is also possible to compare various functions *f(t)* and *g(t)* using the cross-correlation function:

$$\Psi_K(\tau) = \int_{-\infty}^{+\infty} f(t) \times g(t-\tau) dt$$

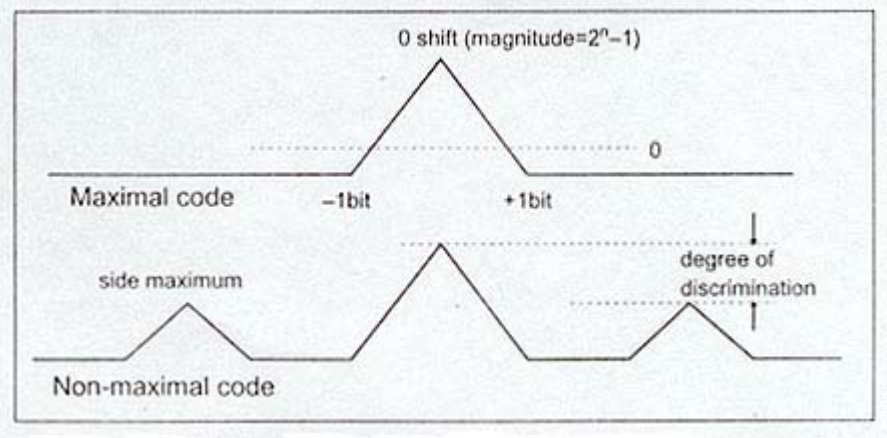
This cross-correlation function is a measure of the degree of agreement between functions. Since the functions to be compared are different $\Psi_K(\tau)$ may never achieve the maximum value of $\Psi_A(\tau)$. It is an indication that the functions are different when a certain threshold (-1 in the case of a binary) is not exceeded.

In the correlation of binary code sequences, the result for cross-correlation will be +1 if the functional values coincide and -1 if they do not. The integration then forms a summing of all bits of the code. The correlation value for a certain phase-shift can therefore be simply calculated by placing the bits over another and comparing them bit by bit. The correlation rate is the sum of agreements and disagreements.

For example, the maximal code sequence 1110010 is compared with itself in the seven possible phase-shifts.

shift	sequence	agreements	disgreements	agreements minus disagreements
0	1110010	-	-	-
1	0111001	3	4	-1
2	1011100	3	4	-1
3	0101110	3	4	-1
4	0010111	3	4	-1
5	1001011	3	4	-1
6	1100101	3	4	-1
7	1110010	7	0	+7

As can be seen the auto-correlation function value is always -1, except for the case of coincidence, where it is a maximum. The greater the length of the code, the higher the auto-correlation amplitude and the greater the code discrimination or cross-correlation response. The auto-correlation function for maximal and non-maximal codes are shown in the drawing below. As shown in the figure, maximal codes have only one auto-correlation maximum, whereas non-maximal codes have side maxima as well.

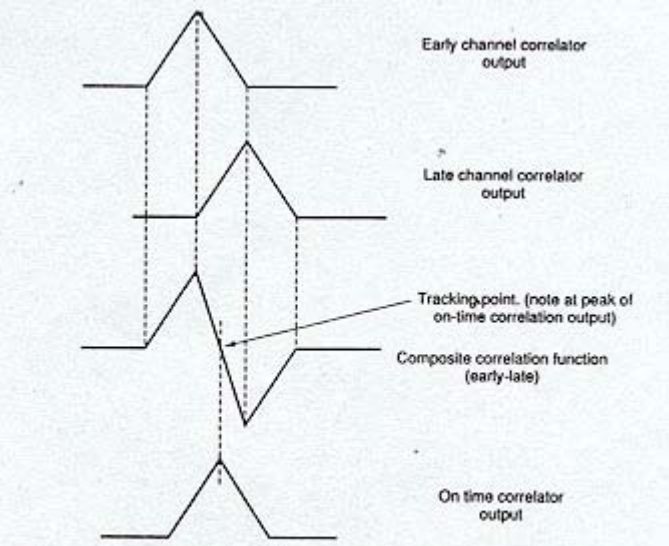


When non-maximal codes are used it is important to ensure that a sufficiently large spacing exists between the main and side maxima. Despite these disadvantages, non-maximal codes are used to exploit their main advantages of rapid synchronisation and message security.

8.4 The Delay Locked Loop

After initial acquisition, the spread spectrum receiver must maintain synchronisation by tracking changes in the transmitter's PN code clock. The circuitry required is known as a tracking loop, as it tracks the transmitter's code clock frequency variations. Without a tracking loop synchronisation will be lost as the transmitter and receiver PN code clocks will tend to drift apart.

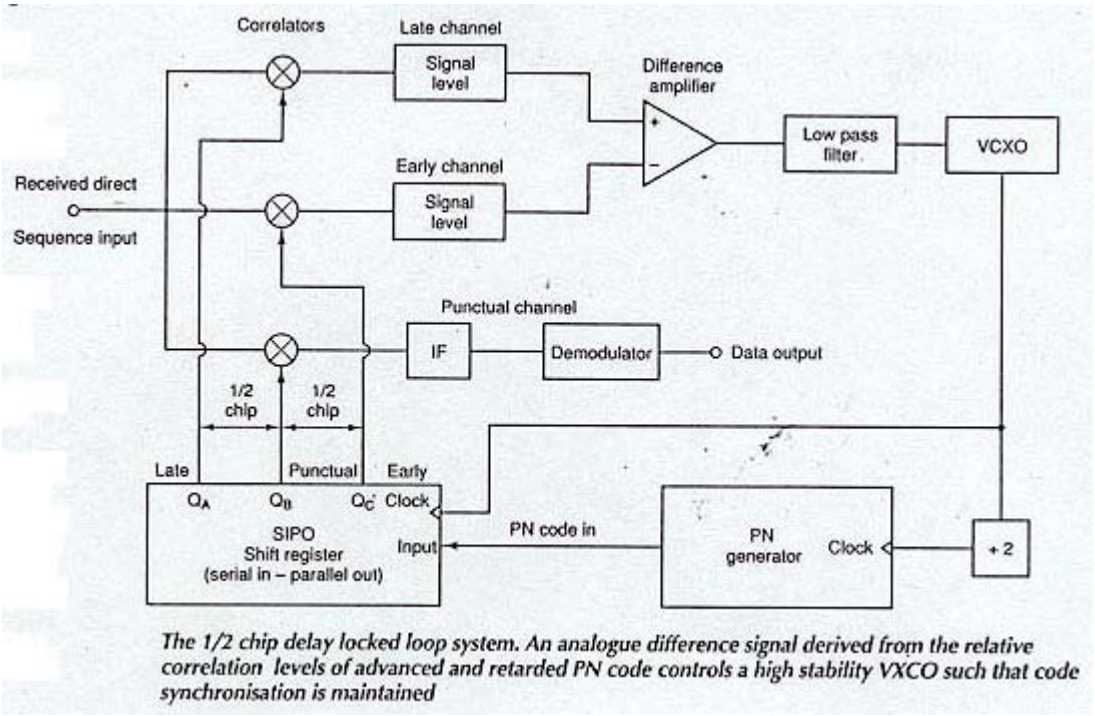
In a delay locked loop two identical pseudo-random or PN despreading codes are delayed with respect to each other. Each PN code is used in separate correlators (early and late) to despread (correlate) the received direct sequence signal. The result of correlation between an incoming direct sequence signal and the receiver PN code is a triangular function two chips (code bits) wide. Assuming synchronisation two correlated signals (each with a triangular correlation waveform) are produced with their correlation peaks separated by the delay between the early and late receiver PN codes. If the two correlation signals are summed in a difference amplifier and filtered, then a composite correlation function is produced. This composite correlation function has a linear region between its maximum and minimum values.



The composite correlation function and correlator waveforms in the 1/2 chip delay locked loop.

If this composite correlation function is used to control the receiver's code clock frequency (for example by driving a voltage controlled oscillator) then the receiver will track the transmitter's code clock at a point halfway between the maximum and minimum values of the composite correlation function.

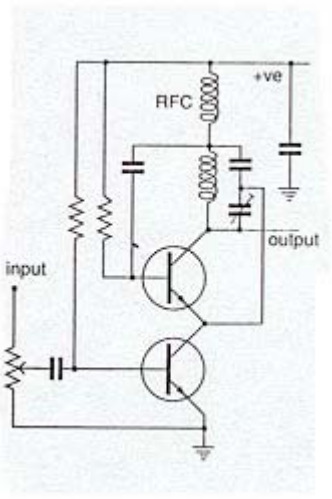
An optimum solution is to have a third on-time (punctual) PN sequence correlator channel for signal recovery, with early and late correlators simply providing tracking to keep the on-time channel in the middle of the correlation window. Such an approach provides an optimally correlated (despread) output signal for subsequent data demodulation.



The 1/2 chip delay locked loop system. An analogue difference signal derived from the relative correlation levels of advanced and retarded PN code controls a high stability VXCO such that code synchronisation is maintained

8.5 The Synchronous Oscillator

The synchronous oscillator is an elegant but little known circuit which can be used to advantage where a phase-locked loop (PLL) would normally be employed. The SO is a free-running oscillator which oscillates at a frequency determined by its LC tank with no signal applied to its input.



When a signal is applied within the SO's acquisition bandwidth the oscillator synchronises and tracks the input signal. The SO output amplitude is constant when locked to and tracking an input signal. A decrease in the input carrier-to-noise ratio reduces the SO's tracking bandwidth to maintain a constant signal-to-noise ratio at the SO's output. This characteristic allows a SO to acquire and track very noisy signals.

The SO can also act as a frequency multiplier or divider. In the direct sequence receiver, the SO locks to a noisy 12 MHz signal and provides a stable 6 MHz output. This function could be achieved using a PLL but the SO has many advantages[5,6,7,8] and, as it is based on only two transistors, is much simpler to implement.

A simplistic explanation of the SO operation is that the upper transistor acts as a Class C oscillator. The upper transistor only conducts for a very brief period of time; when the upper transistor conducts, there is a voltage across the lower transistor biasing it allowing it to conduct. At this time the input signal can then be injected to synchronise the oscillator. During the rest of the oscillator cycle input noise is unable to enter the oscillator as the lower transistor is reverse biased. This arrangement produces coherent amplification which is why the SO can extract signals from very low signal-to-noise inputs.

TUF1, RK3 and MAR series devices are manufactured by Mini-Circuits (from Cirkit in the UK and Dale Electronics, Camberley) FX609J from Consumer Microcircuits, Witham, Essex, England. Crystals are available from IQD Ltd, Crewkerne, Somerset, England. RFFM2 DBM comes from Walmore Electronics, London, England.

References

1. *Convert NRZ to Biphase*. John T. McGaughey. *Electronic Design* April 12 1990

2. *High Performance 70cm Pre-Amp*. Timothy Edwards. *Radio and Electronics World* March 1982

3. Appendix 4 - *Multiplication of Direct Sequence Signals*. *Spread Spectrum Systems* Second Edition Robert C. Dixon. Wiley Interscience.

4. *Communication in the Presence of Noise*. C.E. Shannon. *Proc.IRE*, vol.37, No1,(pp10-21) Jan 1949.

5. *The Synchronous Oscillator: A Synchronisation and Tracking Network*. Vasil Uzunoglu & Marvin H. White. *IEEE Journal of Solid-State Circuits*. Vol. SC-20, No.6 December 1985.

6. *Synchronous and the Coherent Phase-Locked Synchronous Oscillators: New Techniques in Synchronisation and Tracking*. Vasil Uzunoglu & Marvin H. White. *IEEE Transactions on Circuits and Systems*. Vol.36, No7, July 1989.

7. *A Practical Direct-Sequence Spread Spectrum UHF Link*. Andre Kesteloot, N4ICK. *QST magazine*. [Re-printed in the ARRL Spread Spectrum Handbook (pp 8-47 - 8-54) The American Radio Relay League.].

8. United States Patents: 4 335 404, 4 274 067 and 4 356 456.

Bibliography

Spread Spectrum Systems Second Edition. Robert C Dixon, Wiley Interscience, ISBN 0-471-88309-3.

The ARRL Spread Spectrum Sourcebook. Editors Andre, Kesteloot N4ICK and Charles Hutchinson K8CH. The American Radio Relay League. ISBN 0-87259-317-7.

Spread Spectrum Communications Volume 1, 2 and 3. Marvin K Simon, Jim K Omura, Robert A Scholtz and Barry K Levitt. Computer Science Press. ISBN 0-88175-017-7 (Set).

Coherent Spread Spectrum Systems Jack K Holmes. Wiley Interscience. ISBN 0-471-03301-4.

Digital Communication by Satellite. James J. Spilker Jr. Prentice-Hall. ISBN 0-13-214155-8.

---

9 Appendix B – Bill of material

Transmitter schematics

Qty	Value	Device	Package	Distributor	Order number	Parts
1	1M	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R1
1	100	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R2
1	4MHz	CRYTAL AT cut , 8MHZ	18U-V	Farnell	103-878	XT1
1	5-50p	variable capacitor		Farnell		C2
1	40p	disk plate capacitor	2.54 mm grid	college store		C1
1	74HC4040N	74HC4040N , binary counter	DIP16	Farnell	380-880	U2
1	74HC14N	74HC14N , shmitt trigger	DIP14	Farnell	380-404	U3
1	74HC74N	74HC74N , Dtype flip-flop	DIP14	Farnell	380-477	U4
1	74HC86N	74HC86N, XOR gate	DIP14	Farnell	380-490	U5, U7, U9
1	74HC08N	74HC08N , AND gate	DIP14	Farnell	380-374	U8
1	74HC164N	74HC164N, 8bit shift register	DIP14	Farnell	380-623	U6
1	PUSH_BUTTON	PUSH_BUTTON		Farnell	150-542	SW1
1		SLIDE_SWITCH		Farnell	674-345	SW2
1	1.5u	capacitor		college store		C3
1	TUF-1	TUF-1 , balanced mixer	see datasheet	Minicircuits UK	see letter	M1
2		BNC to PCB connector		Farnell	583-560	CN1, CN2
1	100n	disk plate capacitor	5.08 mm grid	college store		C4
2	10	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R13, R14
1	150	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R15
1	33k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R8
1	3.3k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R9
due software problems 10n disc plate decoupling capacitors are missing , but they are implemented on board						

Despreading schematics

Qty	Value	Device	Package	Distributor	Order number	Parts
2		BNC to PCB connector		Farnell	583-560	X1, X2
3	3.3k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R5, R17, R29
12	10	SMD resistor	1206	Farnell	912-232	R1, R2, R10, R11, R13, R14, R22, R23, R25, R26, R34, R35
12	10n	SMD capacitor	1206	Farnell	499-353	C1, C2, C3, C6, C13, C15, C16, C25, C27, C28, C29, C38
3	15u	tantal capacitor	EUTAP5-50	Farnell	643-580	C11, C19, C32
4	24	SMD resistor	0805	Farnell	321-7863	R9, R21, R33, R37
1	74HC86N	74HC86N, XOR gate	DIP14	Farnell	380-490	IC2
3	100k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R7, R19, R31
22	100n	disk plate capacitor	5.08 mm grid	college store		C7, C8, C9, C10, C12, C14, C18, C20, C21, C22, C23, C24, C26, C31, C33, C34, C35, C36, C37, C39, C41, C42
7	100n	SMD capacitor	1206	Farnell	499-389	C4, C5, C17, C30, C40, C43, C44
6	150	SMD resistor	1206	Farnell	912-372	R3, R12, R15, R24, R27, R36
3	560	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R8, R20, R32
3	600ohm@100MHz	SMD inductor (RF suppresor)	0805	Farnell	305-6569	L1, L2, L3
6	680	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R4, R6, R16, R18, R28, R30
3		MAR-6 amplifier	see datasheet	Minicircuits UK	see letter	A1, A2, A3
1	PIF-70	PIF-70 , 70MHz band pass	see datasheet	Minicircuits UK	see letter	F1
3	SA605D	SA605 integrated IF system	SOL20	Farnell	302-6309	IC1, IC3, IC5
6	SFE6.0	SFE6.0 ceramic filter	see datasheet	GES,Prague, Czech rep.		FIL1, FIL2, FIL3, FIL4, FIL5, FIL6
3	TUF-1	TUF-1 , balanced mixer	see datasheet	Minicircuits UK	see letter	M1, M2, M3

diphase schematics

Qty	Value	Device	Package	Distributor	Order number	Parts
2	600ohm@100MHz	SMD inductor (RF suppresor)	0805	Farnell	305-6569	L1, L2
1		LED5MM	5mm grid	college store		U$1
1	1.5k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R9
1	1M	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R12
3	1k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R2, R13, R14
1	1n	disk plate capacitor	2.54 mm grid	college store		C13
1	2k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R3
2	3.3k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R1, R8
10	10n	disk plate capacitor	2.54 mm grid	college store		C1, C2, C3, C4, C5, C6, C7, C8, C9, C12
2	10u	tantal capacitor	EUTAP5-50	Farnell	643-683	C16, C17
1	22k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R6
1	50k	trimr		college store		R7
2	74HC74N	74HC74N , Dtype flip-flop	DIP14	Farnell	380-477	IC3, IC4
2	74HC86N	74HC86N, XOR gate	DIP14	Farnell	380-490	IC1, IC2
2	74HC221N	74HC221N, monostable	DIP16	Farnell	117-423	IC5, IC6
1	74HC4046A	74HC4046A , digital PLL	DIP16	Farnell	380-891	U1
1	100k	trimr		college store		R11
2	100n	disk plate capacitor	5.08 mm grid	college store		C10, C11
1	100n	SMD capacitor	1206	Farnell	499-389	C15
1	100p	disk plate capacitor	2.54 mm grid	college store		C14
1	220	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R5
1	220p	disk plate capacitor	2.54 mm grid	college store		C18
1	560	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R4
1	LM311N	LM311N , comparator	DIP14	college store		IC7

DLL schematics

Qty	Value	Device	Package	Distributor	Order number	Parts
1		SLIDE_SWITCH		Farnell	674-345	S2
2	1.5u	capacitor		college store		C1, C5
1	1M	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R13
1	1n	disk plate capacitor	2.54 mm grid	college store		C2
1	3.3k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R19
1	8MHz	CRYTAL AT cut , 8MHZ	18U-V	Farnell	103-914	Q1
1	10k	trimr		college store		R17
8	10n	disk plate capacitor	2.54 mm grid	college store		C4, C6, C7, C8, C9, C10, C11, C12
1	22p	disk plate capacitor	2.54 mm grid	college store		C3
2	33k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R6, R18
1	47	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R14
1	47k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R10
1	50k	trimr		college store		R5
1	74HC04N	74HC04N , inverter	DIP14	Farnell	380-362	IC3
1	74HC08N	74HC08N , AND gate	DIP14	Farnell	380-374	IC6
1	74HC14N	74HC14N, Smitt trigger	DIP14	Farnell	380-404	IC8
1	74HC86N	74HC86N, XOR gate	DIP14	Farnell	380-490	IC7
2	74HC164N	74HC164N, 8bit shift register	DIP14	Farnell	380-623	IC4, IC5
1	74HC4040N	74HC4040N , binary counter	DIP16	Farnell	380-880	V7
10	100k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R1, R2, R3, R4, R7, R8, R9, R11, R15, R16
1	390k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R12
1	BB809	BB809 , varicap diode		GES,Prague,Czech rep.		D1
1	LM324N	LM324N , opamp	DIP14	college store		IC1
1	PUSH_BUTTON	PUSH_BUTTON		Farnell	150-542	S1
1	uA741P	uA741P	DIP8	college store		IC2

Demonstartion circuit schematic

Qty	Value	Device	Package	Distributor	Order number	Parts
2		LED5MM	5mm grid	college store		U$1, U$2
1	2.2u	capacitor		college store		C4
3	3.3k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R2, R6, R8
1	5.6k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R1
1	10k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R3
5	10n	disk plate capacitor	2.54 mm grid	college store		C2, C3, C5, C6, C7
1	47k	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R7
1	74HC14N	74HC14N, Smitt trigger	DIP14	Farnell	380-404	IC1
1	74HC86N	74HC86N, XOR gate	DIP14	Farnell	380-490	IC2
1	74HC221N	74HC221N, monostable	DIP16	Farnell	117-423	IC3
1	100p	disk plate capacitor	2.54 mm grid	college store		C1
2	220	standart resistor, tolerance 5%	0207/ 10mm grid	college store		R4, R5
1	PUSH_BUTTON2	push button		college store		B1

Created in 2002-2003 as DLX4 graduating project on Cork Institute of Technology, Ireland

Author: Daniel Kraus
Erasmus exchange program student
Home institution: Czech Technical University in Prague, Faculty of Electronic Engineering
Email: dankr@jikos.cz